Cross-Layer Analysis for Video Transmission over COFDM-based Wireless Local Area Networks

Pierre Ferré

January 2006

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Engineering, Department of Electrical and Electronic Engineering.

Word Count: 59,348

Abstract

In recent years, Wireless LANs have been successfully integrated into a range of consumer electronic products. Thanks to new technologies and low cost radio devices, the trend is now toward high speed wireless transmissions and in many cases, WLANs are now about to replace traditional wired LANs. Providing services such as multimedia communications, video on demand and real-time live broadcast distributed onto TVs, PCs or PDAs is becoming a reality.

The purpose of this thesis is to investigate a cross-layer analysis in order to provide reliable video transmission. The COFDM-based PHY layer of IEEE 802.11a/g at 2.4 and 5.2GHz is examined together with the IEEE 802.11 MAC. Emphasis is placed on the poor MAC throughput efficiency and its asynchronous access using CSMA/CA and retransmission. The MAC generates variable delay at the application layer. The study has focused on the new H.264 video coding standard which provides good compression efficiency, a suite of error resilience tools and a 'network friendly' approach.

In order to reduce the strong reliance on the mandatory ARQ mechanism in the IEEE 802.11 MAC layer, enhancements to the PHY layer are proposed, using combined Space-Time Block Codes and Reed Solomon Coding. PER improvements of up to 10dB are demonstrated and the number of retransmissions is reduced.

Video packetisation strategies are investigated in order to overcome the poor throughput efficiency of the IEEE 802.11 MAC without losing video quality. The proposed modifications to the legacy MAC allow the recovery of error-free NAL units in a corrupted MAC frame when several NAL units are mapped into one single MAC frame. Results are presented for the particular case of live broadcast transmission to several handheld devices and a gain of 3dB in PSNR is obtained for a PER or 3×10^{-2}. An interleaving cross-packet FEC is also studied and a gain of 6dB in PSNR is obtained with a coding rate of 0.875 and a depth of 8 packets for a PER of 10^{-2}.

In WLANs, traditional link adaptation algorithms use retransmission to optimise the error-free data throughput. These methods do not take into account the time bounded nature of the video data. In this thesis, a link adaptation algorithm based on video quality is presented. Rather than using the PSNR as a switching metric, the use of PER thresholds is investigated. Empirical simulations shows that the PER thresholds used for throughput-based algorithms are too high for low latency video applications. To maintain good video quality, lower thresholds are necessary and results show a gain of 7dB in PSNR. The influence of parameters such as the number of retransmissions, packet length, video bit rate, content and concealment techniques is discussed.

Finally, real measurement data is collected with WLANs at 2.4GHz and 5.2GHz. Logged data is used to simulate video transmission and to investigate the influence of packet length on the received video quality. The acquired data is also used to stress the need for smart video packetisation strategies. Moreover, results show a gain of 5dB in PSNR when an appropriate link adaptation algorithm designed for time bounded video transmission is used.

Je n'ai déjà pas assez de temps pour tout faire.
Alors comment voulez-vous que je fasse le reste?
Le Chat

Acknowledgements

First of all, I would like to thank Professor David Bull and Professor Andrew Nix for their continuous enthusiastic supervision. They provided invaluable support in every aspect of my research.
I would also like to thank Dr Angela Doufexi for the help, guidance and advice she provided on Wireless Networks and for answering countless technical questions. I would also like to thank Dr James Chung-How and Dr Dimitris Agrafiotis for their help and useful advice on Video Coding and Transmission. I would like to thank both my supervisors, Dimitris, James and Angela for reviewing this manuscript and for providing me with useful feedback.
I also wish to thank Tuan Kiang Chew for the very interesting discussions we had on the WLAN measurement. I would also like to thank to ProVision Communications Limited, Bristol, UK, for their support.
I would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council (EPSRC), UK and ProVision Communications Limited, Bristol, UK. Without this funding, this work would not have been possible.
Un grand merci à toute ma famille et à tous mes amis, Stef, Nadège, Nico, Nino, Armelle, Pauline et tous les autres pour le bout de brousse que nous avons fait, que nous faisons et que nous ferons ensemble. C'est un peu grâce à eux que je suis ce que je suis. Et bien sur, j'ai aussi une pensé pour mes amis de Bristol (Xavier, Aymeric, Claudia, Monica, Muge, Damien, Erwin entre autres) qui ont rendu cette vie bristolienne bien agréable. Et pour finir, une énorme merci à Carolina pour m'avoir supporté, aidé, soutenu et bien plus encore durant ces 4 années.

Author's Declaration

I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original excepts where indicated by special references in the text and no part of the dissertation has been submitted for any other degree.
Any views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.
The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

DATE: 31/01/06

SIGNED: 

Copyright

List of Publications

1. Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix and David Bull, *Link Adaptation for Video Transmission over COFDM-based Wireless LANs*, IEEE Symposium on Communications and Vehicular Technology (SCVT), October 2003, Eindhoven, ***Best Paper Award Winner.***

2. Pierre Ferré, Angela Doufexi, Andrew Nix and David Bull, *Throughput Analysis of the IEEE 802.11 and IEEE 802.11e MAC*, IEEE Wireless Communications and Networking Conference, March 2004, Atlanta.

3. Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix and David Bull, *Enhanced Video Streaming over COFDM based Wireless LANs using combined Space Time Block Coding and Reed Solomon Concatenated Coding*, IEEE Vehicular Technology Conference (VTC), May 2004, Milan.

4. Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix and David Bull, *Packetisation Strategies for Enhanced Video Transmission over Wireless LANs*, IEEE Packet Video workshop (PV), December 2004, Irvine.

5. T.K Chiew, Pierre Ferré, Dimitris Agrafiotis, Araceli Molina, Andrew Nix and David Bull, *Cross Layer WLAN Measurement and Link Analysis for Low Latency Error Resilient Wireless Video Transmission*, IEEE International Conference on Consumer Electronics, January 2005, Las Vegas

6. Dimitris Agrafiotis, T.K Chiew, Pierre Ferré, David Bull and Andrew Nix, *Seamless Wireless Networking for Video Surveillance Applications*, Invited Paper, SPIE, Electronic Imaging, January 2005, San Jose

7. Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix and David Bull, *Robust Video Transmission over Wireless LANs*, Submitted to IEEE Transactions on Vehicular Technology (VT), second submission under review.

8. Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix and David Bull, *Video Quality Based Link Adaptation for low latency video transmission over Wireless LANs*, to be submitted to IEEE Packet Video Workshop (PV), June 2006, China.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

AARF:	Adaptive Auto Rate Fullback
AC:	Access Category
ACF:	Association Control Frame
ACK:	Acknowledgment
AEC:	Advanced Error Concealment
AFC:	Adaptation Field Control
AIFS:	Arbitrary Inter Frame Spacing
AP:	Access Point
ARF:	Auto Rate Fallback
ARIB:	Association of Radio Industries and Businesses
ARQ:	Automatic Repeat reQuest
AVC:	Advance Video Coding
AWGN:	Additive White Gaussian Noise
BER:	Bit Error Rate
BMA:	Boundary Match Algorithm
BMME:	Block Matching Motion Estimation
BPSK:	Binary Phase Shift Keying
BO:	Back Off
C/N:	Carrier To Noise Ratio
CABAC:	Context-Adaptive Binary Arithmetic Coding
CAVLC:	Context-Adaptive Variable Length Coding
CBP:	Coding Block Pattern
CCIR:	International Radio Consultative Committee (Former ITU-R)
CF:	Contention Free
CFB:	Contention Free Burst
CFP:	Contention Free Period
CIF:	Common Intermediate Format
CL:	Convergence Layer
COFDM:	Coded OFDM
CP:	Contention Period
CRC:	Cyclic Redundancy Check
CSI:	Channel State Information
CSRC:	Contributing Source
CTS:	Clear To Send
CSMA/CA:	Carrier Sense Multiple Access / Carrier Avoidance
CW:	Contention Window
DCC:	DLC user Connection Control
DCF:	Distributed Coordination Function
DCT:	Discrete Cosine Transform
DFD:	Displaced Frame Difference
DIFS:	Distributed Inter Frame Spacing
DLC:	Data Link Control
DON:	Decoding Order Number
DPCM:	Differential Pulse Code Modulation
DVB:	Digital Video Broadcasting
DVB-S:	Digital Video Broadcasting - Satellite
DVB-T:	Digital Video Broadcasting - Terrestrial

Eb/No:	Energy To Noise Density Ratio
EC:	Error Control
EDCA:	Enhanced Distributed Channel Access
EDCF:	Enhanced Distributed Coordination Function
ERC:	European Radio communication Committee
EREC:	Error Resilient Entropy Coding
ES:	Elementary Stream
ETSI - BRAN:	European Telecommunication Standards Institute - Broadband Radio Access Network
FC:	Frame Control
FCC:	Federal Communication Commission
FCS:	Frame Check Sum
FEC:	Forward Error Correction
FFT:	Fast Fourier Transform
FMO:	Flexible Macroblock Ordering
FU:	Fragmentation Unit
GI:	Guard Interval
GoB:	Group of Blocks
GoP:	Group of Pictures
HC:	Hybrid Coordinator
HCCA:	Hybrid Coordination Channel Access
HCF:	Hybrid Coordination Function
Hiperlan/2:	HIgh PErformance Radio Local Area Network Type 2
HiSWANa:	High-Speed Wireless Access Network Type a
HVS:	Human Vision System
IEEE:	Institute of Electrical and Electronic Engineers
RM:	Resynchronisation Marker
DCT:	Inverse Discrete Cosine Transform
IDR:	Instantaneous Decoder Refresh
IEC:	International Electrotechnical Commission
IFFT	Inverse Fast Fourier Transform
IP:	Internet Protocol
IPv4:	Internet Protocol Version 4
IPv6:	Internet Protocol Version 6
ISI:	Inter Symbol Interference
ISO:	International Organisation for Standardisation
ISM:	Industrial Scientific Medical
ITU:	International Telecommunication Union
LAN:	Local Area Network
LOS:	Line of Sight
MAC:	Medium Access Control
MB:	Macroblock
MC:	Motion Compensation
ME:	Motion Estimation
MPEG:	Motion Picture Experts Group
MRRC:	Maximum-Ratio Receiver Combining
MSE:	Mean Square Error
MT:	Mobile Terminal

MTAP:	Multi-Time Aggregation Packet
MV:	Motion Vector
NACK:	Negative Acknowledgement
NAL:	Network Abstraction Layer
NAV:	Network Allocation Vector
NALU:	Network Abstraction Layer Unit
NER:	NAL unit Error Rate
NLOS:	Non Line Of Sight
OAR:	Opportunistic Auto Rate
OFDM:	Orthogonal Frequency Division Multiplex
OSI:	Open System Interconnection
PAC:	Packet Concatenation
PC:	Point Coordinator
PCF:	Point Coordination Function
PCMCIA:	Personal Computer Memory Card International Association
PDA:	Personal Digital Assistant
PDU:	Protocol Data Unit
PER:	Packet Error Rate
PES:	Packet Elementary Stream
PFC:	Previous Frame Copy
PHY:	Physical
PIFS:	Point coordination Inter Frame Spacing
PLCP:	Physical Layer Convergence Procedure
PMD:	Physical Media Dependent
PPDU:	PLCP Protocol Data Unit
PS:	Program Stream
PSC:	Prefix Start Code
PSNR	Peak Signal to Noise Ratio
QAM:	Quadrate Amplitude Modulation
QCIF:	Quarter CIF
QoS	Quality of Service
QP:	Quantiser Parameter
QPSK:	Quaternary Phase Shift Keying
R-BAR:	Receiver-Based Auto Rate
RBSP:	Raw Byte Sequence Payload
RCC:	Radio Resource Control
RLC:	Radio Link Control
RMS:	Root Mean Square
RPS:	Reference Picture Selection
RS:	Reed Solomon
RSSI:	Receive Signal Strength Information
RTP:	Real-Time Transport Protocol
RTS:	Ready To Send
RVLC:	Reversible VLC
SIFS:	Short Inter Frame Spacing
SDU:	Service Data Unit
SEI:	Supplemental Enhancement Information
SIF:	Source Input Format

SISO:	Single Input Single Output
SN:	Sequence Number
SNR:	Signal to Noise Ratio
SODB:	String Of Raw Data Bits
SSRC:	Synchronisation Source
STAP:	Single-Time Aggregation Packet
STBC:	Space Time Block Codes
TBTT:	Target Beacon Transition Time
TC:	Traffic Categories
TCP:	Transport Control Protocol
TDD:	Time Division Duplex
TDMA:	Time Division Multiple Access
TPC:	Transmit Power Control
TS:	Transport Stream
TSPEC:	Traffic Specification
TXOP:	Transmission Opportunity
UDP:	User Datagram Protocol
UP:	User Priority
U-NII:	Unlicensed National Information Infrastructure
VCEG:	Video Coding Experts Group
VCL:	Video Coding Layer
VLC:	Variable Length Code
VLSI:	Very Large Scale Integration
WLAN:	Wireless Local Area Network

Chapter 1

Introduction

1.1 Context of the research

In recent years, portable computers, hand-held terminals, laptops, personal digital assistants (PDAs) and mobile phones have flourished in everyday life while mobile communications have experienced massive commercial growth [1]. Initially used as professional tools or reserved for privileged sectors of the population, with the advent of low-cost and reliable technologies, this equipment is now widely available to the entire population. Wireless solutions are rapidly replacing traditional wired systems.

More specifically, Wireless Local Areas Networks (WLANs), providing wireless connectivity, were first installed into companies and corporate environments to enable the sharing of information. The market is now targeting a wider population by offering ease of installation, connectivity, flexibility and mobility. WLANs are already deployed in some public places, such as shops and cafes, where customers can have wireless access to the Internet via laptops and PDAs. In addition to flexibility and connectivity, the growing trend is toward faster and faster links. For the last decade, Wireless LANs standardised by the Institute of Electrical and Electronic Engineering (IEEE) at 2.4GHz (802.11b [2] and 802.11g [3]) and at 5.2GHz (802.11a [4]) in USA, and WLANs standardised by ETSI-BRAN at 5.2GHz (Hiperlan/1 [5] and Hiperlan/2 [6]) in Europe, have been at the centre of academic and commercial research. With new technologies such as Orthogonal Frequency Division Multiplexing (OFDM) at the physical layer (PHY), bit rates have increased up to 54 Mbits/s. The IEEE 802.11 Task Group N, which is still at the proposal stage, is now aiming at rates over 100 Mbits/s. Thanks to their mobility, these emerging networks, offering high speed wireless transmission, are now competing with traditional wired Ethernet LANs or optical links that offer 100-150 Mbits/s [7].

Today, real-time video, interactivity and high definition are becoming the customer requirements in terms of multimedia services. Table 1.1 [7] shows an estimate of the bit rates required for various multimedia services. They typically require large bandwidths, especially when high quality is needed. The peak bit rate is typically around 10 Mbits/s for a single user when all the services are combined [7]. Digital Versatile Disk (DVD) streams may however require up to 30 Mbits/s when functionality such as fast

forward is used. Previous wireless systems such as the Universal Mobile Telecommunications System (UMTS) which supports 2 Mbits/s do not provide sufficient bandwidth to support such services. UMTS may however be sufficient for low bit rate multimedia with a low resolution. With fast links of up to 54 Mbits/s already offered by emerging WLANs, real-time video transmission and interactive multimedia applications are about to become a wireless reality [8].

Table 1.1: Typical Application Bit rates for Multimedia Services

Services	*Requirements*	*Bit rate*
Voice/Audio	Low Delay	8 - 256 kbits/s
Digital/Audio	Low Error	0.1 - 10 Mbits/s
Video Telephony (H.263)	Low Error	64 - 384 kbits/s
Motion Video (MPEG-2)	Low Delay	1.5 - 10 Mbits/s
HDTV	Low Delay	up to 20Mbits/s

1.2 Existing research

Video transmission has been the focus of much research in recent years and technologies supporting such transmission are now widely available. Wired systems were first investigated, especially with the deployment of IP-based (Internet Protocol) [9] networks. Many resilient encoding and robust transmission techniques have been developed in order to overcome problems arising from a IP-based transmission on wired networks. In [10], the authors review video coding techniques and error resilient options for video streaming. Moreover, problems encountered during the video delivery are presented and are related to bandwidth variation, delay and loss. Error resilient techniques offering robustness for video transmission over error-prone channels have been widely researched and developed in the literature [11, 12, 13, 14]. Similarly, error concealment techniques using various features of the encoded video to recover missing macroblocks at the video decoder have been investigated [15, 16]. Mitigating the effect of losses and networks bandwidth variations via adaptation techniques [17, 18, 19, 20, 21] have also been studied in great detail. In parallel, Wireless LANs in the IEEE 802.11 family have been studied at the Medium Access Control (MAC) level [22, 23, 24, 25] as well as at the physical (PHY) level with Coded OFDM-based (COFDM) IEEE 802.11a [4] and IEEE 802.11g [3]. These earlier studies failed to acknowledge the nature of the transmitted data [26, 27, 28, 29], with many treatments assuming that video is simply another form of data.

For Internet-like transmission over a wired network, best effort methods are traditionally considered [30] for video streaming. However, more recently, with the advent of new wireless technologies, research has considered the streaming of video over wireless channels, taking into account the nature of the channel [31]. Digital Video Broadcasting Terrestrial (DVB-T [32]) and Digital Video Broadcasting for Handheld devices (DVB-H [33]) are now available for streaming video over wireless networks. Both use

a robust COFDM-based transmission. However, these systems only support broadcast transmission of Motion Picture Expert Group type 2 (MPEG-2) video and are connection orientated. Moreover, they do not provide any flexibility and support mainly video services. Services such as high speed Internet access and distribution are not initially supported.

Research groups have mainly focused on particular aspects of the wireless/video space and have rarely interacted with other domains. The Technical University of Berlin (TU-B) [30], Technical University of Munich (TU-M) [34] and the Heinrich Hertz Institute in Berlin [35] have been particularly active in the video coding area and especially the latest H.264 video coding standard [36]. The Multimedia Wireless Networking group at the Seoul National University [22], the ComNets group at Aachen University in Germany [23], and the Philips Research group [37, 38] have been intensively researching on the IEEE 802.11 Medium Access Control Layer (MAC) [39] and its latest enhancement, IEEE 802.11e, supporting Quality of Service [40]. Physical layer (PHY), and especially OFDM and various Multiple Input Multiple Output (MIMO) systems and signal processing techniques have been widely investigated by many research groups around the world. For example, the Mobile Communications department of EURECOM, France [41], the Centre for Wireless Communications at the University of Oulu, Finland [42], the Wireless System lab at Stanford University, USA [43], are extensively researching the PHY transmission and signal processing fields, such as Space Time Block Codes, MIMO, Mutli Carrier Code Division Multiple Access (MC-CDMA), channel estimation and equalisation. The University of Bristol, in the UK, is deeply involved in the area of the PHY layer, including OFDM, especially OFDM based WLANs [44], MIMO [45] and propagation [46, 47]. Many European Union FP6 projects are investigating wireless physical layer solutions for radio transmission, such as the WINNER project (Wireless World Initiative New Radio), which aims to develop new concepts in radio access with improvements in data rate, latency, spectrum efficiency, and coverage.

Few groups have however actively combined these areas. The TEMICS group of IRISA in France is focusing on joint source channel coding for video transmission over wireless channels. Adaptive multimedia transmission protocols are investigated at INRIA Sophia-Antipolis in France [48] with a network-centric point of view. The Mobile Communications Department of EURECOM in France has just started to investigate cross-layered WLAN optimisation [41, 49] for multimedia transmission. The University of California, Davis USA [50], in collaboration with Philips Research, has actively participated in the integration of video over IEEE 802.11-based networks, and in particular the enhanced IEEE 802.11e version. The Technical University of Munich (TU-M), with the latest research in H.264, has been been involved in the development of H.264 mutlimedia transmission over current and future cellular networks, such as 3G, UMTS or GSM [34]. Stanford University, USA, is investigating adaptive network oriented techniques to allocate resources [51, 52]. Their research has also focused on channel-adaptive video streaming [53], in order to provide efficient, robust, scalable and low-latency video

transmission, via rate-distortion and packet scheduling with feedback information [54]. Ongoing work at Stanford University also includes the transfer of information between layers to adapt various parameters [55], the incorporation of video packet loss due to routing and network congestion for a model of video streaming [56]

The WINHOME (Wireless INnovation in the HOME) EU ESPRIT Project 25048 (1998-2000) aimed to provide high quality interactive television (MPEG-2), Internet access, videophone communications (via existing satellite and cable Set-Top Boxes or other Multi-Media Server equipment), using a flexible home distribution platform based on HIPERLAN products [7, 57]. The transmission of MPEG-2 video over Hiperlan/2 has been studied in [58] with the use of error resilient video transcoding as a means of handling channel errors. The Open Infotainment Services In Radio Interconnected Systems (OSIRIS) project developed as part of the 3C Research program, UK, aims to develop novel technologies to provide a pervasive, scalable and resilient radio infrastructure, thus allowing seamless access to multimedia services and content via WLAN [59]. A key task of this project is the Quality of Service (QoS) provision for a wide range of wireless audio/visual multimedia applications. The Wireless Cameras and Audio-Visual SeAMless Networking (WCAM) project developed by the Information Society of the European Commission (EU FP6 IST-2003-507204 [60] 2003-2005) is currently studying and a validating a system for audio-visual content delivery over a wireless, seamless and secured network. One aspect of this project is the distribution of secured multimedia over WLANs. The Wi-Fi Alliance [61] is currently developing a certification (Wi-Fi Multimedia (WMM) certification) [62] for QoS support for multimedia applications over Wi-Fi. This is mainly assessing QoS issues such as traffic prioritisation and shared and scheduled access to the medium among different applications.

Video streaming over WLANs has however yet to reach the same level of in-depth study as its wired or 3G/UMTS counterparts. In particular, interactions and cooperations between the various layers has not yet been studied in depth. The 'Wireless' world and the 'Video' world have rarely been jointly investigated and an analysis carried across the layers needs to be performed in order to provide high quality and network efficient video transmission.

1.3 Motivations of the research

Technological advances now allow high capacity wireless data transmission with mobility, large bandwidths and flexibility. Moreover, the various peripherals deployed in a house (TVs, PCs, video and DVD players, laptops, PDAs, satellite and cable receivers) are now available at low prices. Installing a wireless LAN in such an environment would permit a seamless connection of these peripherals into one wireless multimedia network as shown in figure 1.1.

Such WLAN systems are, of course, not only limited to home environment where peripherals are either interconnected via an access point or directly connected in an ad-hoc manner for the distribution of video and multimedia content. They can also be deployed into universities, institutions, companies and other corporate environments

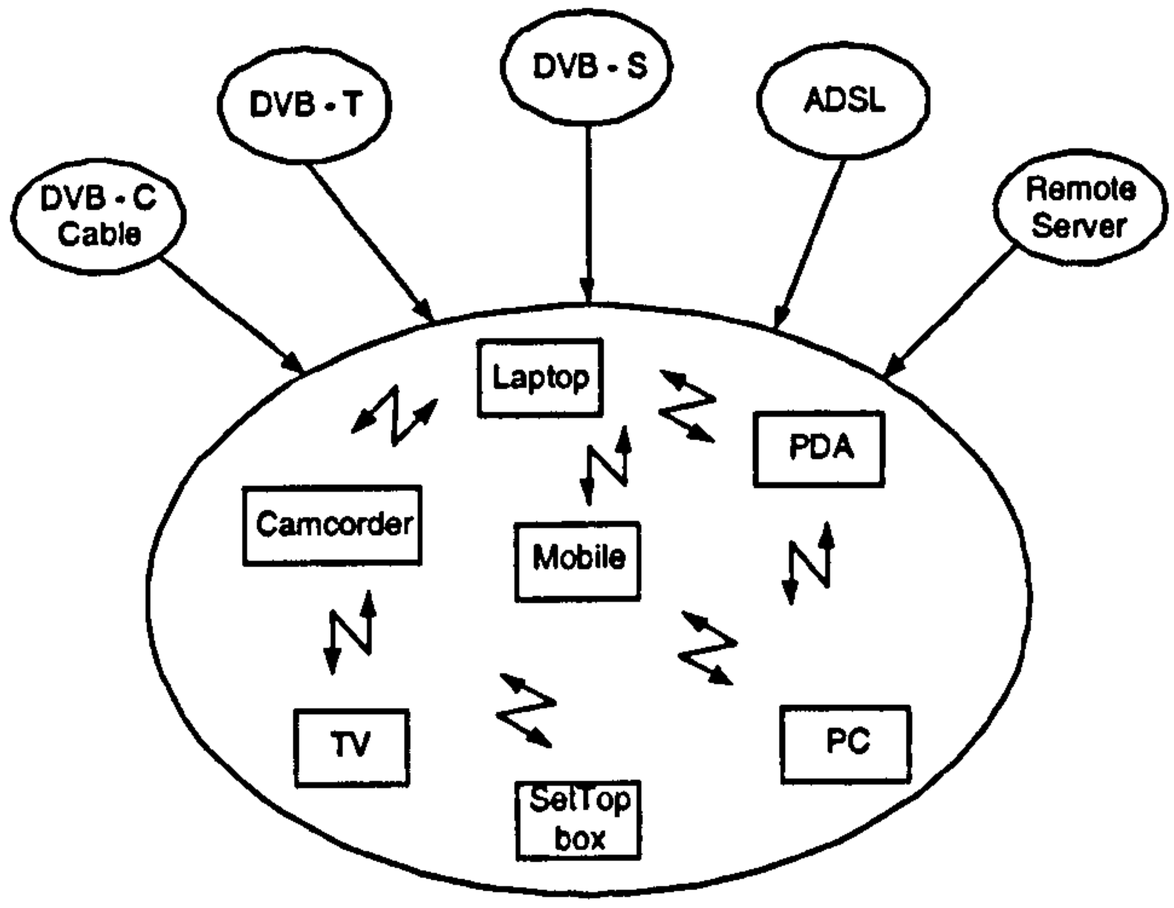


Figure 1.1: Structure of an indoor seamless connection via WLANs

for applications where multimedia and interactivity are key requirements. The range of applications for such a design is very wide and includes:

- Distribution of real-time home video, broadcast, home multimedia and entertainment with a SetTop Box with high video quality
- Distribution of real-time video, broadcast, onto PDAs, PCs, laptops
- Video on demand and multimedia access: distribution of pre-encoded video to laptops, PCs, TV sets or PDAs, such as in hotel environments
- High speed Internet access and distribution around the house
- Wireless video projectors
- Wireless camcorders within a wireless surveillance networks
- Wireless control of electronic devices such as DVD, VCR players or lights.
- Audio mp3/wave playback

For both personal use (private consumers) and public use (corporate environments), the demand for high quality real-time multimedia interactivity with mobility is real and strong. On the other hand, the technology of WLANs with Orthogonal Frequency Division Multiplex (OFDM) at 2.4 and 5.2GHz is reaching a point where it is now possible to answer these demands: mobility, flexibility, data throughput and capacity. All the ingredients are therefore present for the implementation of a high speed multimedia network using a new generation of Coded OFDM-based WLANs at 2.4 and 5.2GHz. Because of its very strong sensitivity to error and delay, one of the most challenging tasks when designing such a wireless system is real-time video transmission.

In addition, as explained in section 1.2, most of the research to date has been carried out on a single layer basis, where each is studied on its own without knowledge

of the other layers. There is need for further research where layers inter-operate and coordinate their efforts to enhance the overall performance of the system.

1.4 Research Approach

Integrating multimedia services, and in particular real-time video transmission into a seamless wireless network raises many issues. Previous studies have mainly focused on enhancing specific components of the transmission, rather than considering the system as a whole. Applying the International Standards Organisation (ISO) reference model as shown in figure 1.2 [63], these issues can be addressed at each layer. Note that with the Open Systems Interconnection (OSI) environments, the three upper layers, namely *Application*, *Presentation* and *Session* are considered as being application-orientated and can be considered as a single entity [63]. Every layer in the stack of figure 1.2 uses its own mechanisms and parameters and these can be improved to enhance performance. However, implementing a system similar to the one depicted in figure 1.1 by simply concatenating layers (from the application to the physical layer) with already optimised and available devices and drivers would not lead to optimum system performance. Each of these parameters and mechanisms affect the overall performance of the system. For example, modifying one parameter at the application layer might affect the performance of another layer and therefore the whole system. Similarly, modifying one parameter at the physical layer might affect the performance of other layers. Their influence should therefore not only be understood at each layer, but also at a system level. Since all these parameters and mechanisms are closely linked, it is not be convenient to address these issues on a single layer basis, but instead, the system should be understood as a whole. An analysis that covers all the layers is preferable, where each layer is not only understood separately but also interacts and cooperates with the other layers. Understanding a mechanism in a specific layer might be useful to select and design mechanisms at another layer. More specifically, layers may coordinate their effort and share information in order to provide the highest Quality of Service (QoS). In the context of this thesis, the received video quality at the application layer and the delay/jitter at the MAC layer are the key QoS parameters to optimise. This *Cross-Layer* approach is applied in this thesis in order to understand which parameters influence the performance of a video stream sent over WLANs.

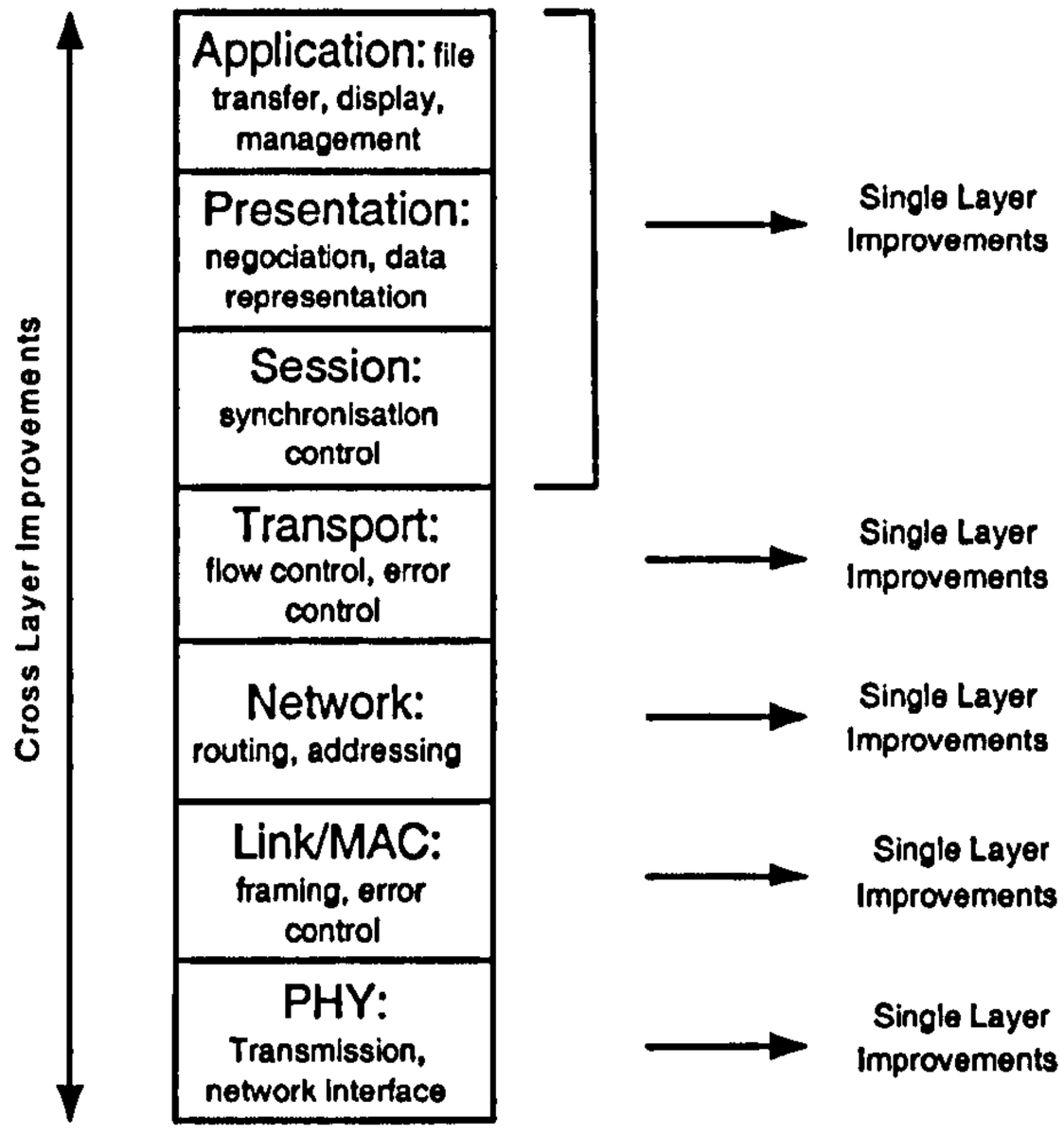


Figure 1.2: ISO Reference Model

1.5 Identification of key issues

Since it requires very large bandwidths, video is compressed and encoded in order to facilitate storage and transmission. This compression is unfortunately lossy and the original video can not be recovered without some loss of quality. Moreover, because of the temporal and spatial prediction coding mechanisms, encoded video is very sensitive to errors. One way of combating errors is to implement a robust and error-resilient encoder that will minimise their effect. The way errors are concealed at the decoder is also critical.

Even with resilient encoding and strong concealment techniques, the state of the link is still critical. Because of its wireless nature, the channel does not offer reliable performance. The PHY implements various components in order to combat channel errors. Once the quality of the link is assessed, specific mechanisms at the PHY may be required to ensure a better link. These mechanisms can include enhancements to the PHY by adding diversity, such as MIMO, or by adding Forward Error Correction (FEC) for error recovery.

In addition, wireless systems need to provide fair access to the medium for all users, creating therefore the possibility for an asynchronous transmission. This is the role of the MAC layer. The way the overall system performance is affected by the access mechanism needs to be understood since real-time video requires a time-bounded transmission, with high data rates needed if high definition is used. Delay and jitter need to be minimised and throughput optimised. The various delay mechanisms therefore need to be identified, as well as the parameters influencing the throughput. Note that layers above may implement Automatic Repeat reQuest (ARQ) in order to overcome

the PHY's relatively poor performance, even though ARQ is not desirable for real-time video streaming. Broadcast transmission does not allow any retransmissions and thus represents the most challenging form of wireless video transmission.

As stated earlier, new emerging wireless LANs such as IEEE 802.11a/g and Hiperlan/2 provide high link speeds up to 54 Mbits/s. This rate is, however, calculated at the PHY layer and the data rate available at the application layer is reduced due to various overheads. It is therefore important to assess the different parameters that will affect the available bit rate at the upper layers. Channel errors and channel access are known to strongly influence this throughput and their impact also needs to be addressed. More particularly, the way the video packets from the encoder are mapped onto the lower layers is not trivial. A simple encapsulation might not be an optimal solution and packetisation strategies must be investigated.

IEEE 802.11a/g and Hiperlan/2 provide multiple operating modes with different characteristics (modulation, coding rate, bit rate). The algorithm that determines which mode will be selected is crucial and will play a large role in determining system performance. The different parameters taken into account for the modulation and coding mode selection need to be studied in order to provide the best transmission for video. The metric chosen for the mode selection may include PHY-orientated, network-oriented or application-orientated parameters. The content of the transmitted data, the bit rate, the received quality and the throughput should be considered, confirming the cross-layer nature of the research requirement.

IEEE 802.11 b/g cards are nowadays easily available at low cost. Implementations are proprietary and vary according to manufacturer. Performance may therefore vary from one brand to another. Moreover, the behaviour of a specific chip might not be appropriate for video transmission.

1.6 Aims of the research

The main aim of this thesis is to devise methods for the efficient and robust transmission of video, using a cross layer analysis, The approach focuses on the new encoding video standard H.264 [36] in conjunction with COFDM-based WLANs at 2.4 and 5.2 GHz ([3, 4]). H.264 has been chosen for its good compression efficiency as well as for its network friendly approach. The PHY layers have been chosen for their high speed link (up to 54 Mbits/s). The quality of the physical link has first to be assessed. It stands on the top of the IEEE 802.11 MAC, whose access mechanisms and behaviour need to be investigated in order to highlight problems and drawbacks, in terms of delay and throughput, for time bounded applications such as multimedia streaming. Moreover, as it provides error resilience, the robust encoding features available in H.264 need to be considered.

Once all the key parameters of each of these layers have been determined, the aim is to understand, in a cross layer manner, how they affect the system quality in the context of video streaming and how the system can be optimised. Parameters affecting the packet loss at the application layer and the end to end delay need to be identified.

More specifically, the quality of the radio link clearly influences the application loss rate and has an impact on the use of mandatory IEEE 802.11 MAC ARQ and therefore on the delay. Additional coding schemes at the PHY layer could then be required in order to minimise the dependency on ARQ by improving the radio link quality.

As mentioned in section 1.5, the packetisation process from video packets to PHY packets is not trivial. In the context of an IP-based system using the User Datagram Protocol (UDP) [64], along with various aggregation/fragmentation mechanisms that already co-exist, the encapsulation process needs to be investigated and optimised so that it can overcome the IEEE 802.11 MAC throughput drawbacks highlighted previously.

The operating mode selection at the PHY is usually called *Link Adaptation* or *Rate Adaptation* and is not standardised. Implementations and algorithms are left up to manufacturers. Traditionally, algorithms have been designed so that they optimise error-free data throughput. However, this might not be appropriate for video transmission. Link adaption algorithms need to be investigated to see how the overall system performance is influenced. They need to be designed so that they meet the specific and unique requirements of video.

Finally, off the shelf IEEE 802.11b/g PCMCIA cards need to be investigated in terms of PER, delay, bit rate and link adaptation depending on the type of transmission. Ultimately, video transmission needs to be investigated in order to compare measured performance with simulated results.

1.7 Structure of the Thesis

This thesis is organised as follows.

Chapter 2 reviews COFDM-based Wireless Local Area Networks (WLANs) at 5.2 and 2.4GHz, namely Hiperlan/2 and IEEE 802.11a at 5.2GHz and IEEE 802.11g at 2.4GHz. It details the physical layer (PHY) structure and its various components. A description of Orthogonal Frequency Division Multiplexing (OFDM) is given. This chapter also includes PHY layer performance results: Packet Error Rate (PER), Bit Error Rate (BER) and throughput for a Non Line-Of-Sight (N-LOS) office indoor environment with a root mean square (rms) delay spread of 50ns (corresponding to *Channel A* of ETSI-BRAN).

Chapter 3 is dedicated to the understanding of the IEEE 802.11 Medium Access Control (MAC) layer. A detailed description of the various access schemes is given. Since it is the basis for all the other schemes and because it is the only WLAN MAC implemented on available chipsets, emphasis is given to the asynchronous Distributed Coordination Function (DCF) mode and the exponential random back-off mechanism used for the retransmission process. Throughput and delay performances are presented for the case of a 'lightly loaded network', i.e. with a limited number of stations. The influence of packet length and the number of retransmissions is also presented. A study using Markov Chain theory is carried out in order to simulate the various effects of increasing the user number on throughput and delay.

Chapter 4 presents an overview of video coding techniques. It includes a review of the basics of video coding with redundancy reduction. The Moving Picture Experts Group (MPEG) 2 video coding layer is presented, as well as the MPEG-2 system layer which deals with of the packetisation and the multiplexing of video packets coming from the coding layer. This chapter also provides an overview of error resilience and concealment techniques. Error resilience is aimed at providing robustness for a coded video sequence when it has to be transmitted over an error-prone channel. Concealment is aimed at providing error recovery and retrieval of missing data in order to minimise the effect of errors. Chapter 4 describes the latest video standard H.264, developed jointly by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC MPEG group. This new standard not only improves the coding efficiency by adding new features, but it has also been designed for transmission over various and heterogeneous networks. The Network Abstraction Layer (NAL) provides a 'network-friendly' representation of the encoded video so that it is suitable for transmission over existing and future networks. Moreover, H.264 has been developed with transmission over error-prone channels in mind. The standard therefore includes error resilience tools. This chapter contains a description of both the Video Coding Layer (VCL) layer and the NAL, along with the new error robustness features.

These three first chapters provide the basic understanding required for the further and more detailed cross-layer analysis that follows.

Chapter 5 investigates the use of Forward Error Correction (FEC) and Space-Time Block Codes (STBC) at the PHY layer in order to minimise the use of ARQ for MPEG-2 video transmission. First, the IEEE 802.11a PHY is combined with a Reed Solomon (RS) code with different coding ratios (similar to DVB). A description of STBC is then provided. Finally, both RS and STBC are combined with different coding ratios. Results in term of PER, throughput, use of ARQ and delay are presented.

Chapter 6 discusses packetisation issues and proposes strategies to enhance system performance. The way video packets are mapped and transmitted onto the wireless channel has a very strong influence on the throughput and on the robustness of the video. Because of the large MAC overhead and the poor throughput efficiency, small MAC packets lead to low efficiency even with high PHY data rates. Therefore, mapping one video packet into one MAC frame would not be appropriate. On the other hand, small video packets are preferable for error resilience purposes. In order to understand the retransmission and error detection/correction mechanisms available in the protocol stack, the Real-time Transfer Protocol (RTP), the User Datagram Protocol (UDP) and the Internet Protocol are first reviewed. This chapter then studies the mapping of several video packets into one MAC frame. In order to provide robustness and guarantee good video performance, some modifications of the MAC are suggested so that error-free video packets in a corrupted MAC packet can be recovered. Video transmission simulations have been performed using error patterns on a bit basis, generated by an IEEE 802.11a PHY simulator developed previously at the University of Bristol [44]. PSNR, PER and NAL unit Error Rate (NER) improvements over the legacy MAC

are presented with emphasis given to the particular case of broadcast transmission to several handheld devices. In addition, the particular case of a fixed PHY packet length is discussed and the influence of the number of NAL units mapped is studied. Moreover, this chapter also investigates a cross packet FEC with interleaving. PSNR, PER and NER performance results are shown for various FEC coding rates, and various interleaving depths.

Chapter 7 investigates rate adaptation algorithms. Given the numerous possible operating modes of IEEE 802.11a/g, link adaptation algorithms are used to determine the mode that will operate according to specific metrics. A review of existing and possible solutions is first given. These are more network orientated solutions and do not take into account the time bounded nature of the video data. Because encoded video does not respond to errors in a similar way to other data, it should not be transmitted in the same manner. This chapter therefore discusses a possible link adaptation scheme based on enhancing the perceived quality of the received video rather than the link throughput. The natural switching metric is the PSNR, but this can not be easily implemented. Instead, PHY PER is used and a comparison with traditional algorithms is carried out by using error patterns. Similar to the previous chapter, the particular case of transmission with no or limited numbers of retransmission is highlighted. The influence on switching points between the various PHY modes is studied empirically for various parameters, such as content, number of retransmissions, video bit rates, packet length and error resilience techniques. Implementation issues are finally discussed.

Chapter 8 investigates and studies video transmission over real IEEE 802.11a/g wireless LANs. A description of the hardware and the client/server software used is first given. Data is logged by this software and error patterns generated on a packet basis. Measurements are discussed for different scenarios. The influence on the performance of the key parameters highlighted in the previous chapters are studied. A comparison between TCP and UDP links is carried out. The impact of packet length is presented. The packetisation strategies developed in chapter 6 are discussed. Note that the modifications of the MAC could not be implemented due to the lack of flexible hardware. The need to encapsulate is stressed. The proposed link adaptation algorithm of chapter 7 is also investigated. Such an algorithm could not be implemented due to hardware constraints. The potential benefits of the video quality based algorithm over the throughput-based algorithm are demonstrated.

Chapter 9 presents a summary of the work performed, draws the final conclusions, and explores recommendations for possible future work.

Chapter 2

Wireless Local Area Networks

This chapter is organised as follows: section 2.1 gives an introduction to Wireless Local Area Networks including the architecture, spectrum allocation and reference model. The PHY layer of IEEE802.11a and Hiperlan/2 is detailed in section 2.2. This section contains a review of the PHY layer and describes the Orthogonal Frequency Division Multiplex (OFDM) technology underlying the PHY layer. Performance results in terms of Packet Error Rate (PER), Bit Error Rate (BER) and throughput are shown in section 2.3. Finally, section 2.4 concludes this chapter.

2.1 Introduction

Traditionally, computers in companies or individual PCs are interconnected via Local Area Networks (LANs) and workstations have access to shared data and shared applications. Wired LANs are interconnected with cables and have fixed locations. There is currently a trend towards alternative technology: wireless LANs (WLANs) with the speed of wired LANs. Because clients and users tend to be mobile but still want to be connected to the LAN, regardless of their location, wired LANs are not always practical. The market is now moving toward WLANs, which can appear to be a more attractive technology. These new wireless LANs are easier to install since no cable is required, thus they can be moved easily to a new location. WLANs create a flexible data communication system allowing data rates as high as wired LANs [65, 66].

Several WLANs standards have been studied and developed in parallel over the last 5-10 years. In the United States, the Institute of Electrical and Electronic Engineering (IEEE) has created the IEEE 802.11 Task Group to ratify standards for WLANs. Technologies span multiple physical encoding types, frequencies and applications for wireless networking applications, in the similar manner to Ethernet standards [2, 3, 4, 39]. In Europe, the European Telecommunication Standards Institute - Broadband Radio Access Network (ETSI - BRAN) has developed Hiperlan (HIgh PErformance Radion LAN), which includes four standards [6, 67]. In Japan, the Association of Radio Industries and Businesses (ARIB) has developed the High-Speed Wireless Access Network (HiSWAN). The IEEE 802.11 and ETSI-BRAN standards are summarised in table 2.1 [65]. The trend is now towards new high-speed wireless LANs such as

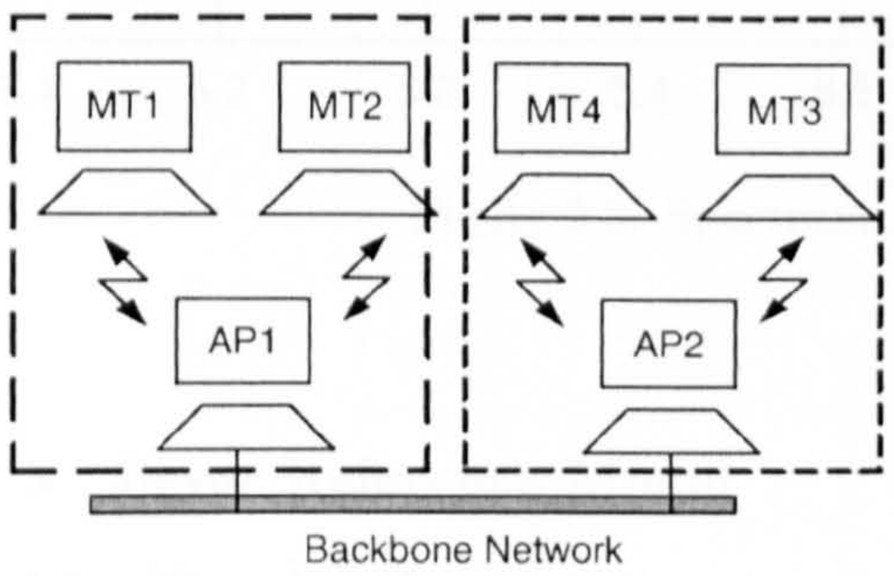
Hiperlan/2 [6], IEEE 802.11a [4] and IEEE 802.11g [3], which allow higher data rates than their preceding WLANs Hiperlan/1 [5] and IEEE 802.11b [2] respectively.

	IEEE 802.11	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g	Hiperlan Type 1	Hiperlan Type 2	Hiperlan Type 3	Hiperlan Type 4
Application	Wireless Ethernet LAN	Wireless ATM	Wireless Ethernet LAN	Wireless ATM	Wireless Ethernet LAN	Wireless ATM	Wireless Local Loop	Wireless Point to point
Frequency Range	2.4GHz	5GHz	2.4GHz	2.4GHz	5GHz	5GHz	5GHz	17GHz
Data Rate	1-2Mbits/s	Up to 54 Mbits/s	5.5 - 11 Mbits/s	Up to 54 Mbits/s	~ 23 Mbits/s	Up to 54 Mbits/s	~ 20 Mbits/s	~ 155 Mbits/s

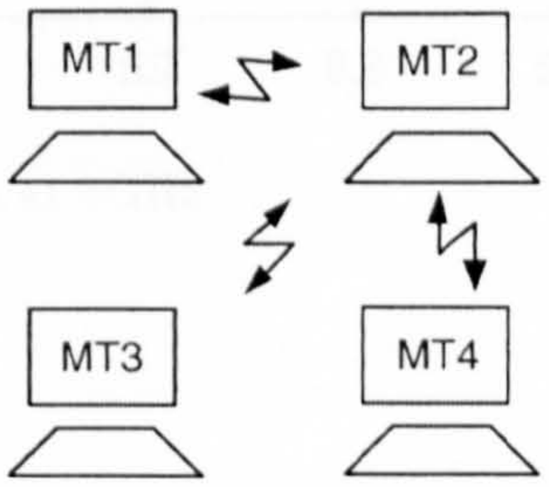
Figure 2.1: Families of Wireless Standards

2.1.1 Architectures and Topologies

Two main types of Wireless LANs may be distinguished: those with a centralised structure (infrastructure) and those without a centralised structure (ad-hoc). The Hiperlan family uses a centralised structure, whereas WLANs from the IEEE802.11 family are deployed in both ad-hoc and infrastructure networks as shown in figure 2.2. In the centralised structure (figure 2.2(a)), the mobile terminal (MT) may communicate with an access point (AP) into a wireless network. MTs may communicate with each other via the AP of their own network. The AP acts as a bridge between the wireless/wired networks and the wireless/wireless network. In an ad-hoc structure (figure 2.2(b)), MTs may communicate directly with each other, without the need for centralised stations.



(a) Centralised Structure with Access Point: Infrastructure



(b) Ah-hoc Structure: Infrastructure

Figure 2.2: Wireless LANs Architectures

2.1.2 Frequency Band and Spectrum Allocation

WLANs are currently working on the 2.4 GHz ISM (Industrial - Scientific - Medical) band. ISM bands include 902-925MHz, 2.4-2.483 GHz and 5.725-5.875GHz. For

the remainder of this thesis, the IEEE802.11a/g [3, 4] and ETSI Hiperlan/2 [6] standards are considered. In the United States, the Federal Communication Commission (FCC) that governs radio transmission has allocated of 300MHz spectrum for unlicenced operation in the 5GHz band: 200MHz from 5.15GHz to 5.35GHz and 100MHz from 5.725MHz to 5.825MHz (U-NII bands). IEEE has developed the 802.11a physical (PHY) layer and the 802.11 Medium Access Control (MAC) layer to operate in these two bands. In Europe, the European Radiocommunication Committee (ERC) has allocated a bandwidth of 455MHz in the 5GHz bands in two parts: from 5.15GHz to 5.35GHz (lower band) and from 5.47GHz to 5.725GHz (upper band), both dedicated to WLANs. ETSI-BRAN has developed its new standard (Hiperlan/2) to operate on these two bands [26, 65, 66]. Figure 2.3 [65, 66] shows the spectrum allocation for Hiperlan/2, HiSWANa and IEEE802.11a [65, 66]. Compared to the 2.4GHz band, the 5GHz is relatively free of interference and by moving to the 5GHz band, performance will increase at the expense of a reduced coverage area. Note that IEEE 802.11g operates in the 2.4GHz band. Even if developed in the US, IEEE 802.11a/g are allowed in Europe with some modifications required for 802.11a.

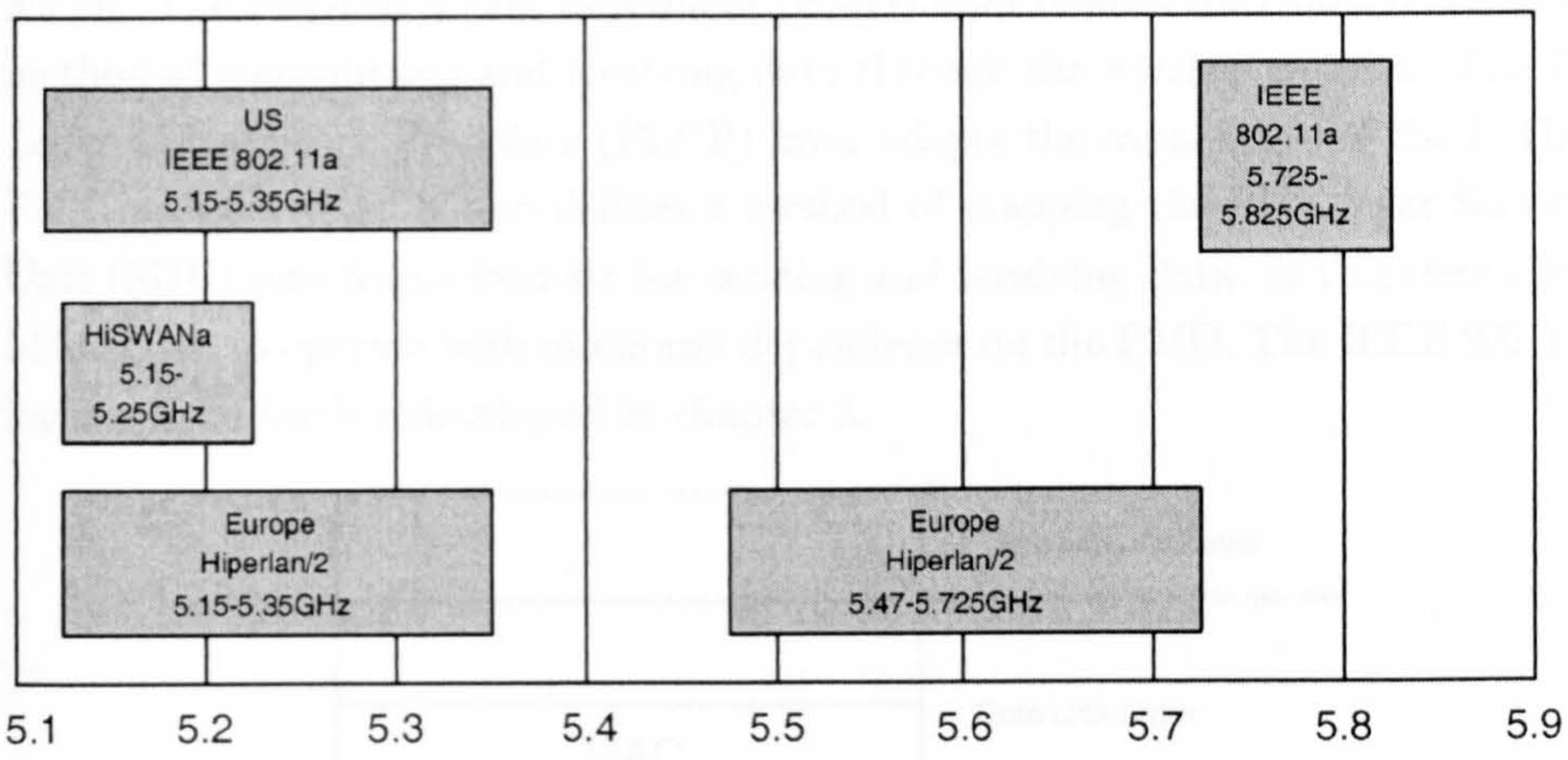


Figure 2.3: Spectrum Allocation at 5GHz

2.1.3 Layer Reference Model

2.1.3.1 Hiperlan/2

The Hiperlan/2 layer reference model is shown in figure 2.4. The Data Link Control (DLC) plane is composed of the Radio Link Control (RLC), the Association Control Frame (ACF), the DLC user Connection Control (DCC), the Error Control (EC) and the Radio Resource Control (RCC) which is in charge of signaling (control messages) between APs and MTs, such as Handover, Dynamic Frequency Selection or Power saving [67, 68].

Since that Hiperlan/2 is not implemented, the focus will be given to the IEEE 802.11 reference model. The IEEE 802.11 family has been initially designed for the

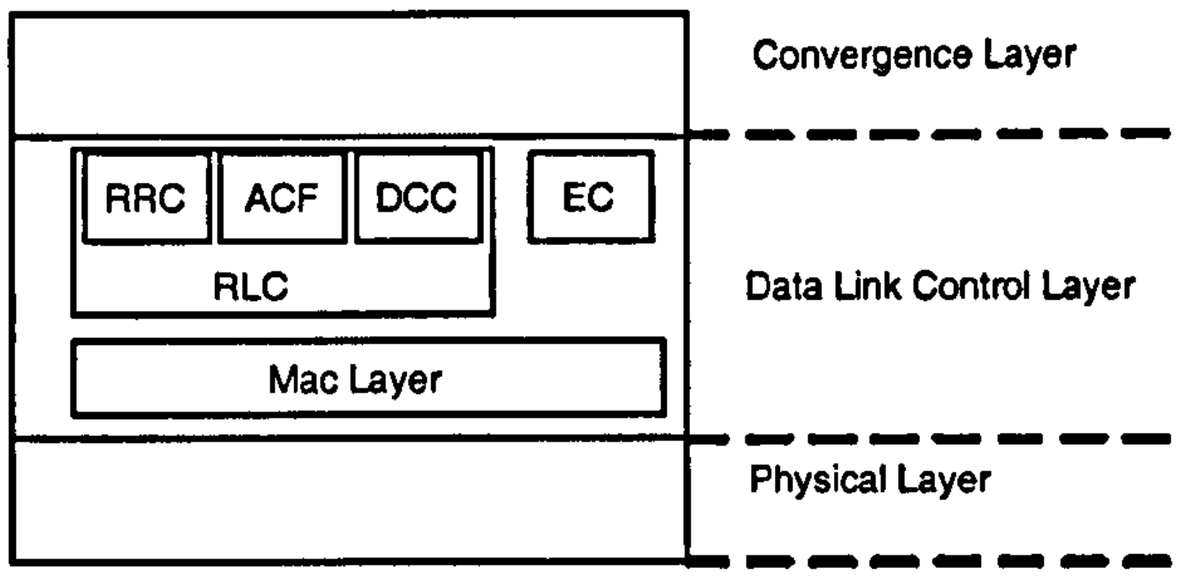


Figure 2.4: Hiperlan/2 layer reference model

'computer' world and has therefore been granted much of the commercial support by manufacturers.

2.1.3.2 IEEE 802.11

The Physical Layer of IEEE802.11a/g is divided into two sub-layers as shown in figure 2.5 [4]. The *Physical Media Dependent* (PMD) layer defines the characteristics and the method of transmitting and receiving data through the wireless channel. The *Physical Layer Convergence Procedure* (PLCP) layer adapts the capabilities of the PMD to the PHY layer services. It also defines a method of mapping the PHY layer Service Data Unit (SDU) into frame formats for sending and receiving data. It therefore allows the MAC layer to operate with minimum dependence on the PMD. The IEEE 802.11 MAC layer will be further developed in chapter 3.

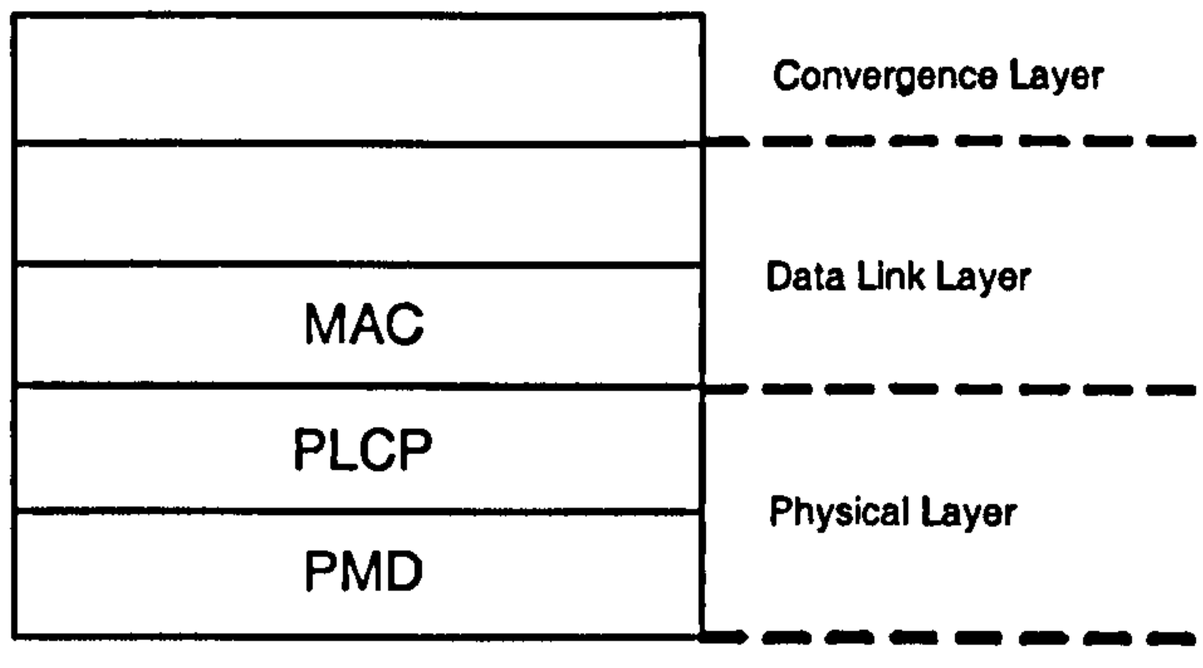


Figure 2.5: IEEE 802.11a/g PHY layer reference model

2.2 COFDM based PHY layer

The physical (PHY) layers defined by Hiperlan/2 [6] and the IEEE802.11a/g [3, 4] standards are nearly identical. The PHY layers of IEEE 802.11a and IEEE 802.11g are identical, the only difference being the operating frequency band (5.2GHz for 802.11a and 2.4GHz for 802.11g). These three WLANs use the same adaptive modulations and coding techniques. IEEE802.11a/g provide 8 operating modes, whereas Hiperlan/2

provides 7 operating modes. Each mode is characterised by its choice of modulation and coding rate. Each operating mode therefore has its own nominal bit rate as shown in table 2.1 [6] [3, 4] [69].

Table 2.1: Mode Dependent Parameters for IEEE802.11a/g and H/2

Mode	*Modulation*	*Coding Rate*	*Nominal bit rate (Mbit/s)*
1	BPSK	1/2	6
2	BPSK	3/4	9
3	QPSK	1/2	12
4	QPSK	3/4	18
5	16QAM (H/2 only)	9/16	27
5	16QAM (IEEE only)	1/2	24
6	16QAM	3/4	36
7	64QAM	3/4	54
8	64QAM (IEEE only)	2/3	48

2.2.1 Physical Transmitter

The physical transmitter for these standards is shown in figure 2.6 [3, 4, 6, 44].

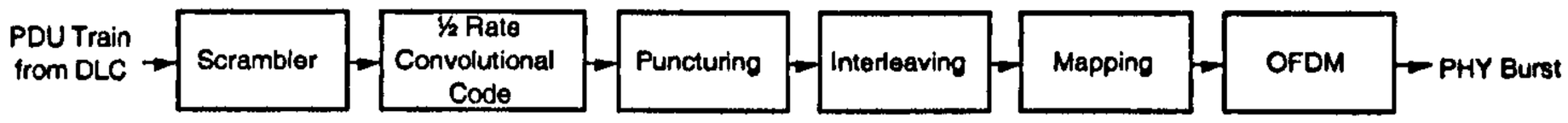


Figure 2.6: IEEE 802.11a/g and Hiperlan/2 Transmitter PHY

The Protocol Data Unit (PDU) train from the DLC layer goes through the following elements:

- A data scrambler in order to prevent long runs of *1*s and *0*s. The three standards use the same length for the pseudo random sequence (127), but they differ in the initialisation of the scrambler. The generator polynomial is $S(X) = X^7 + X^4 + 1$.
- A convolutional encoder with a mother rate of 1/2.
- A puncturer in order to facilitate the use of different coding rates (3/4, 9/16, 2/3).
- An interleaver in order to prevent bursts from being input to the convolutional decoding process at the receiver.
- A mapping process where interleaved data are mapped onto data symbols according to the chosen modulation scheme.
- OFDM, where data symbols are sent on parallel sub-carriers according to the OFDM technique (described in section 2.2.2).

These elements are described in greater details in Appendix A.

2.2.2 Orthogonal Frequency Division Multiplex

Before detailing the use OFDM in the Hiperlan/2 and IEEE 802.11a/g PHY layer, an overview and a mathematical description are given. The set of symbols $A_n e^{j\phi_n}$ at the output of the 'mapper' (see Appendix A.4) is entering the OFDM device.

2.2.2.1 Overview

One of the main problems in wireless transmission is the multipath propagation which induces two phenomena: *Intersymbol Interference* (ISI) and time varying fading [70].

Since a transmitted signal does not arrive at a receiver via single path, but via multiple number of paths, several delayed attenuated and phase shifted replicas signals are received. A symbol can thus be spread or dispersed over time. This leads to ISI, and is defined as *time spreading*. In the frequency domain, time spreading is characterised by the *channel coherence bandwidth*, which defines the range of frequencies over which the channel passes all spectral components with similar gain and linear phase, and hence where frequencies suffer correlated fading [70]. If the bandwidth of the transmit signal is larger than the coherence bandwidth, then the frequencies suffer uncorrelated fading in the frequency domain. Some parts of the signal will experience constructive interference (enhanced signals), whereas some parts may face destructive interference (attenuated signals). This is known as *frequency selective fading*. If the signal bandwidth is smaller than the coherence bandwidth, then all the frequencies are equally affected. This is known as *flat fading* [70]. The time varying nature of the channel can be considered as a *frequency spread* and is characterised by the *Doppler Spread* (or the channel fading rate) [70]. Irreducible errors appear because of the Doppler effect that can not be overcome by simply increasing the Signal to Noise Ratio (SNR).

OFDM is a special form of multi-carrier modulation and was designed to overcome the problems of multipath reception. It transmits a large number of narrow band tones over a wide bandwidth. The basic principle of OFDM is to split a high rate data stream into a number of lower rate streams which are transmitted and modulated simultaneously in parallel over a number of sub-carriers [29, 71]. Only a small amount of data is carried on each sub-carrier and since the data rate on each tone is lower, it reduces the impact of ISI. Sub-carriers are required to be orthogonal. By choosing the correct spacing between sub-carriers, they can be demodulated without any mutual cross-talk and any adjacent carrier interference. The spacing tone is given by $1/OFDM_{period}$, which is equivalent to the Nyquist minimum. OFDM is therefore robust against the adverse effects of multipath propagation and is spectrum efficient [29, 71, 72].

2.2.2.2 Mathematical Description

We recall that two signals φ_p and φ_q are orthogonal if and only if:

$$\int_b^a [\varphi_p \times \varphi_q^*] dt = 0 \tag{2.1}$$

On one sub-carrier, the transmitted signal is the complex wave, $s_c(t) = A_c(t) e^{jw_c t + \phi_c(t)}$,

the real part of which being the real transmitted signal and where $A_c(t)$ and $\phi_c(t)$ are varying on a symbol by symbol basis. w_c represents the sub-carrier frequency. In OFDM, N carriers are considered [29], the complex signal is represented by:

$$s_s(t) = \frac{1}{N} \times \sum_{n=0}^{N-1} A_n(t) e^{j(w_n t + \phi_n(t))} \tag{2.2}$$

It is assumed that carriers are equally spaced ($w_n = w_0 + n\Delta w$). Δw is the sub-carrier spacing and w_0 represents the frequency of the first carrier. On a symbol basis, $A_n(t)$ and $\phi_n(t)$ do not depend on time and are equal to A_n and ϕ_n respectively. The signal is sampled at the sampling rate 1/T over one symbol period τ ($\tau = NT$). The complex signal can be rewritten as:

$$s_s(kT) = \frac{1}{N} \times \sum_{n=0}^{N-1} A_n e^{j((w_0 + n\Delta w)kT + \phi_n)} \tag{2.3}$$

We now assume $w_0 = 0$.

$$s_s(kT) = \frac{1}{N} \times \sum_{n=0}^{N-1} A_n e^{j\phi_n} e^{jn\Delta wkT} \tag{2.4}$$

Equation 2.4 is the *Inverse Discrete Fourier Transform* (IDFT) of $\{A_n e^{j\phi_n}\}$. In the time domain, the signal is $s_s(t)$, and in the frequency domain, it is $\{A_n e^{j\phi_n}\}$. The orthogonality is achieved with $\Delta w = 2\pi\Delta f = \frac{2\pi}{\tau} = \frac{2\pi}{NT}$. OFDM can therefore be implemented with an IDFT or with an *Inverse Fast Fourier Transform* (IFFT) in the case of N being a power of 2. There is no need for special modulators, oscillators and multipliers for each of the subbands[29]. At the receiver, a simple Fast Fourrier Transform (FFT) is performed.

2.2.2.3 OFDM in IEEE802.11a/g and Hiperlan/2

In Hiperlan/2 an IEEE802.11a/g, the stream of complex values $d = (I + jQ) \times K_{MOD} = A_n e^{j\phi_n}$ is divided into groups of 48 complex numbers. These data are modulated onto 48 different carriers, and four others carriers, known as pilots, are used for reference information, and all are transmitted in parallel. The pilot carriers are used to track the reference phase. The 52 sub-carriers compose an OFDM symbol. The 48 sub-carriers used for data are numbered from -26 to +26, excluding 0 which is not used and excluding -21, -7, +7 and +21 which are reserved for the pilots as shown in figure 2.7. Since an IFFT requires the data length to be a power of two, a 64 IFFT is used to perform OFDM on the 52 sub-carriers, where samples that are not modulated are set to 0 as shown in figure 2.8. The use of zero padding is used to create a gap between channels for channel selection filters, anti-alias filters and reconstruction filters. In the frequency domain, the inputs are the complex number output of the 'mapper' and 'pilot inserter': $A_n e^{i\phi_n}$. Coefficients using sub-carrriers numbered from 1 to 26 are mapped to the same numbered IFFT inputs. Coefficients using sub-carrriers numbered from

-26 to -1 are mapped to IFFT inputs 38 to 63. All the others IFFT inputs are set to NULL.

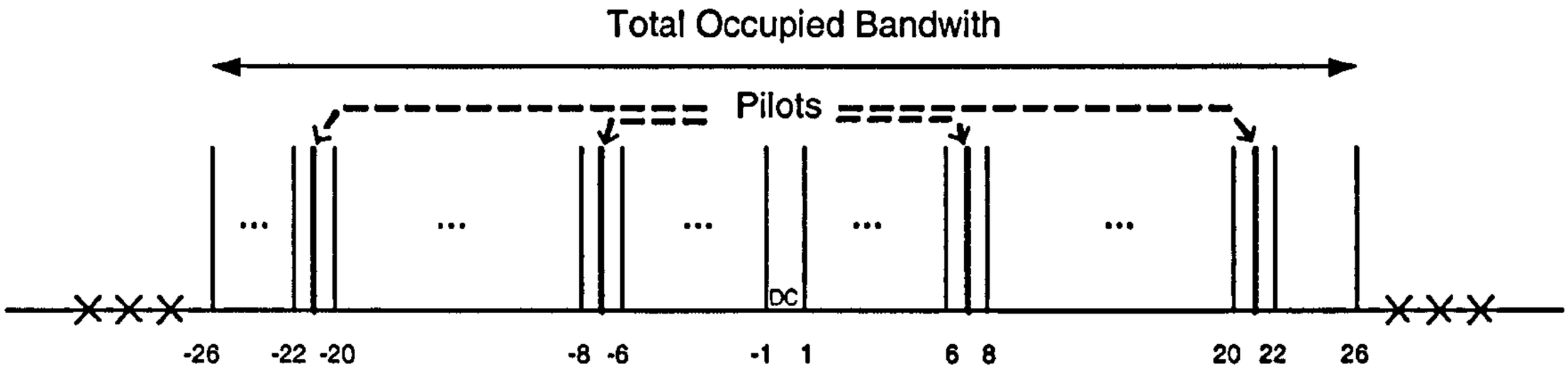


Figure 2.7: Sub-carriers frequency allocation

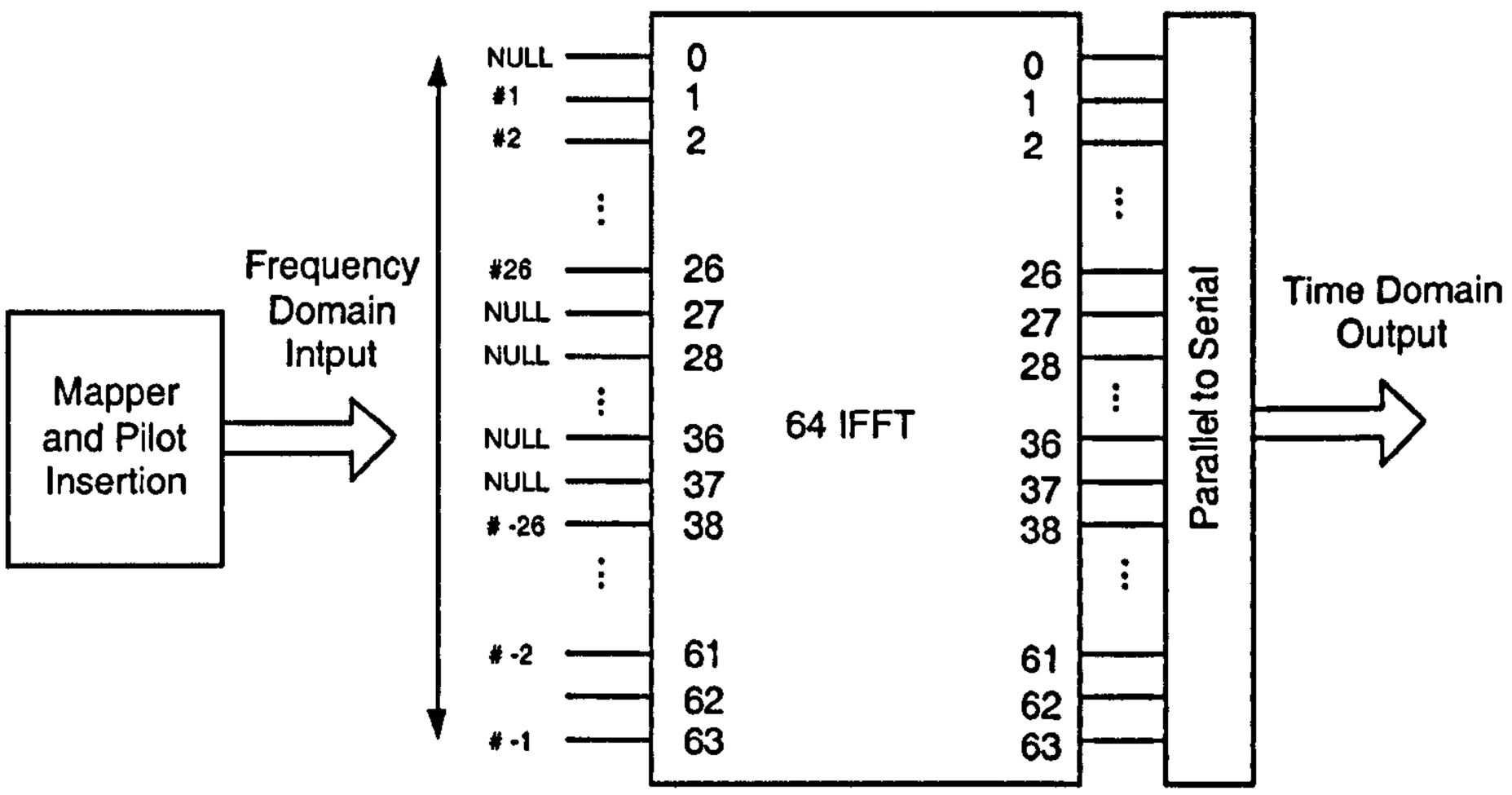


Figure 2.8: OFDM process with a 64 IFFT

One OFDM symbol is transmitted with a duration of T_s. Two parts compose the symbol interval. A useful part with a duration $T_u = \tau$ is followed by a cyclic prefix T_{cp}. The cyclic prefix is known as a *Guard Interval* (GI) and consists of a repeating part of the useful data as shown in figure 2.9. The GI is used to ensure orthogonality in a time spread channel, in addition to helping with timing inaccuracies and synchronisation at the receiver. If a delay echo is short compared to the total symbol period, the energy conveyed by the echo from one symbol to the next one will only affect the GI [29]. The longer the GI, the more rugged the system, but the larger are the overheads and more power and bandwidth is wasted. If the duration of the GI is longer than the excess delay of the channel, then ISI will be eliminated [26].

The length of the useful symbol is equal to 64 samples at a sampling rate of 20 MHz and its duration is $T_u = 64 * \frac{1}{20.10^6} = 3.2\mu s$. The length of the GI is equal to 16 or 8 (optional) samples and its duration is $T_{cp} = 16 * \frac{1}{20.10^6} = 0.8\mu s$ or $0.4\mu s$ for the optional mode. The total transmitted duration is then: $T_s = T_u + T_{cp} = 4\mu s$

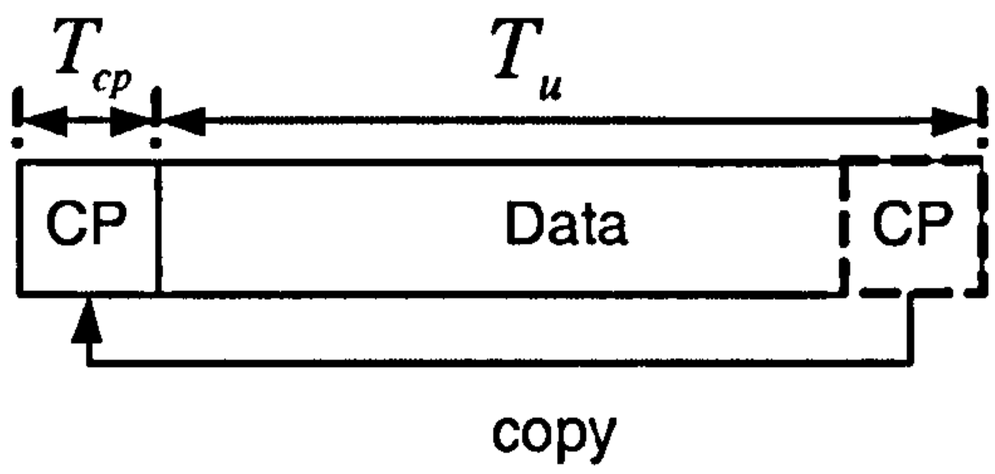


Figure 2.9: Guard Interval of an OFDM symbol

($3.6\mu s$ optional). In order to guarantee the orthogonality of the system, the sub-carrier spacing is chosen such as $\Delta f = \frac{1}{T_u} = 321.5kHz$. The total occupied bandwidth used by an OFDM symbol is then $52 \times 321.5kHz = 16.25MHz$. The OFDM parameters of IEEE802.11a and Hiperlan/2 are summarised in table 2.2 [3, 4, 6]. The baseband format of the transmitted OFDM symbol (output of the 64 IFFT) in the time domain is therefore:

$$s(t) = \sum_{n=-26}^{26} A_n e^{j\phi_n} e^{j2\pi\Delta f(t-T_{CP}-nT_s)} \tag{2.5}$$

providing that $A_0 = 0$ and after cyclic prefix and pilot insertion.

Table 2.2: Numerical values for OFDM parameters in IEEE802.11a/g and Hiperlan/2

Parameters	*Values*
Sampling Rate: $f_s = 1/T$	20MHz
Useful Symbol part duration: T_u	$64 \times T = 3.2\mu s$
Cyclic prefix duration: T_{cp}	$16 \times T = 0.8\mu s$ (mandatory)
	$8 \times T = 0.4\mu s$ (optional)
Symbol Interval: T_s	$80 \times T = 4\mu s$ (mandatory)
	$72 \times T = 3.6\mu s$ (optional)
Short Symbol Interval: T_{Short}	$0.8\mu s$
PLCP Preamble duration: $T_{PLCPpreamble}$	$10 \times T_{short} + 2 \times T_s = 16\mu s$
PLCP Header: $T_{PLCPheader}$	$1 \times T_s = 4\mu s$
Number of data sub-carriers: N_{SD}	48
Number of pilot sub-carriers: N_{SP}	4
Total number of sub-carriers: N_{ST}	52
Subcarrier spacing: Δ_f	0.3125MHz ($1/T_u$)
Spacing between the two outter most sub-carriers	16.25MHz ($N_{ST} \times \Delta_f$)

Table 2.3 summarised the bit allocation per OFDM symbol. For example, QPSK modulation uses 2 bits per symbol in its constellation. 48 carriers therefore carry $48 \times 2 = 96$ coded bits and 48 data bits with a 1/2 rate code (mode 3).

Hiperlan/2 can support 19 non-overlapping channels deployed onto two bands: 8 on the lower band from 5.150GHz to 5.350GHz and 11 on the upper band from 5.470GHz and 5.725GHz [6]. Channels are spaced at 20MHz and the highest PHY rate after

Table 2.3: Bit Allocation for IEEE802.11a/g and H/2

Mode	*Modulation*	*Coding Rate*	*Data Rate in Mbits/s*	*Coded Bits per sub-carrier* N_{BPSC}	*Coded Bits per symbol* N_{CBPS}	*Data Bits per symbol* N_{DBPS}
1	BPSK	1/2	6	1	48	24
2	BPSK	3/4	9	1	48	36
3	QPSK	1/2	12	2	96	48
4	QPSK	3/4	18	2	96	75
5	16QAM (H/2 only)	9/16	27	4	192	108
5	16QAM (IEEE only)	1/2	24	4	192	96
6	16QAM	3/4	36	7	192	144
7	64QAM	3/4	54	6	288	216
8	64QAM (IEEE only)	2/3	48	6	288	192

decoding is 54 Mbits/s. IEEE 802.11a currently supports 12 channels deployed onto three bands: 4 on the lower band from 5.150GHz to 5.250GHz, 4 on the middle band from 5.250GHz and 5.350GHz and 4 on the middle band from 5.725GHz and 5.8250GHz [4]. IEEE 802.11g supports only 3 channels between 2.4GHz and 2.4835GHz.

2.2.3 Physical Burst

The transmitted entity on the PHY layer is called a physical burst (PHY) and is composed of a preamble and payload as shown in figure 2.10. The preamble and the payload consists of OFDM symbols. The baseband format of a PHY burst is: $r_{burst}(t) = r_{preamble}(t) + r_{payload}(t - t_{preamble})$. The time offset $t_{preamble}$ determinates the starting point of the payload section of the burst. Preambles are used for time and frequency synchronisation, for channel estimation and for packet detection.

Figure 2.10: Physical Burst Format

IEEE 802.11a/g transmits PPDU (PLCP PDU) frames which are depicted in figure 2.12 with the corresponding preamble [3, 4]. IEEE802.11a operates on variable length PDUs (up to 4096 bytes) and uses a 4 byte Cyclic Redundancy Check (CRC) code. The header is transmitted with one OFDM symbol at mode 1 (BPSK 1/2 rate). The preamble is composed of 10 short OFDM symbols of $0.8\mu s$ and 2 long OFDM symbols of $4\mu s$ [73]. The preamble is added at the PHY PMD layer. Hiperlan/2 can transmit two types of PDUs, both with fixed size: short PDUs of 9 bytes and long PDUs of 54 bytes, as shown in figure 2.11. The Hiperlan/2 PHY burst is described in more detail in Appendix B.

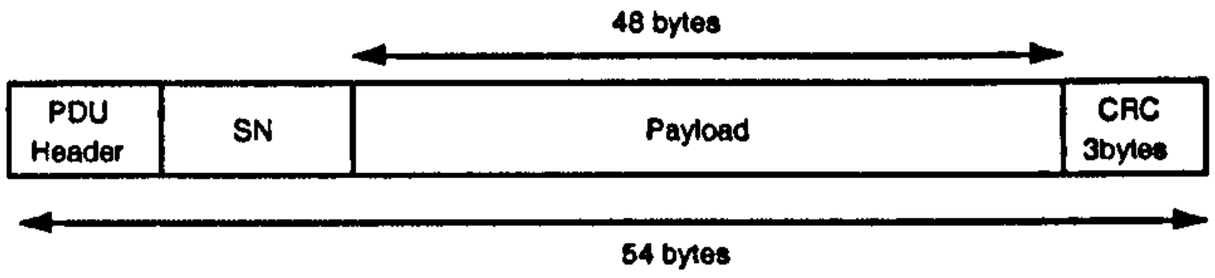


Figure 2.11: Format of a long PDU in Hiperlan/2

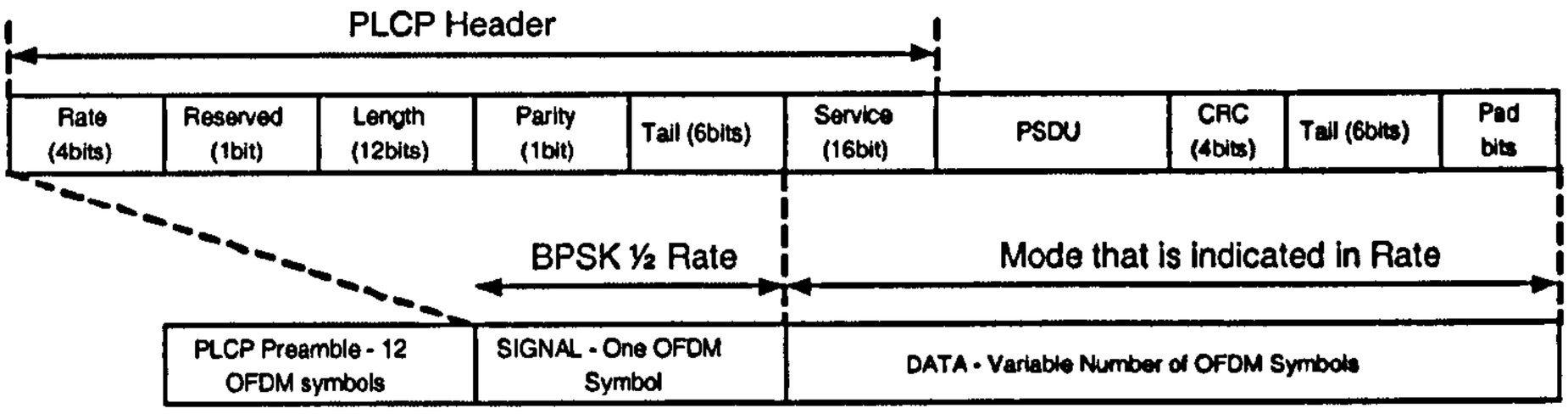


Figure 2.12: PPDU Frame Format in IEEE802.11a

2.3 Performance Results

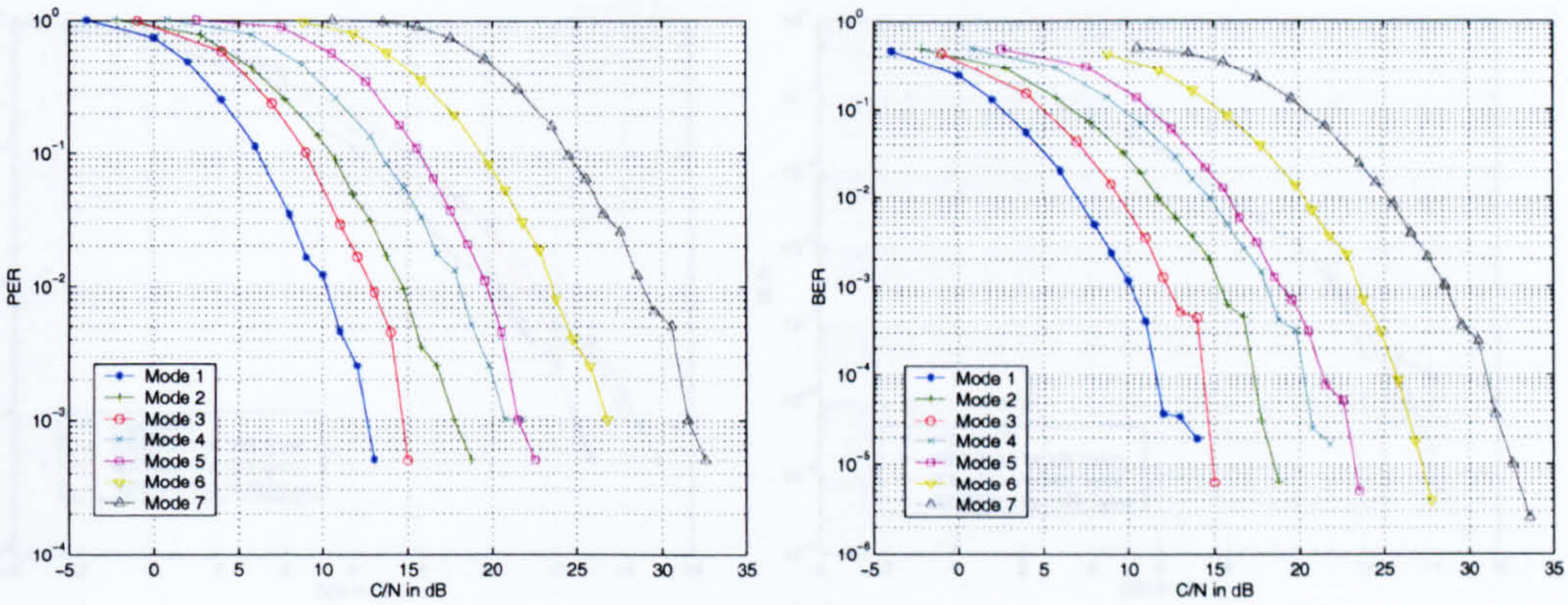
The simulation results presented in this section have been performed using the detailed link-level simulators developed by Dr Angela Doufexi [26, 27] and Dr Michael Butler [44] within the University of Bristol. Simulations have been entirely performed and discussed by the author. Performance results have been obtained for the main 7 modes of Hiperlan/2 and IEEE 802.11a/g PHY layer. The channel model used is the *model A* defined by ETSI-BRAN, and this corresponds to a Non Line of Sight (N-LOS) office environment with 50ns of *rms* delay spread. 2000 uncorrelated wideband channels were generated. Note that IEEE 802.11a and IEEE 802.11g have exactly the same PER, BER and Throughput performance at layer 1, i.e. PHY.

2.3.1 Layer 1 PER and BER Performance

Figure 2.13 shows the PER and BER performance versus Carrier to Noise Ratio (C/N) for Hiperan/2. Even though mode 1 (BPSK 1/2 rate) and mode 3 (QPSK 1/2 rate) have the same E_b/N_o requirements, mode 1 provides better performance than mode 3. Similarly, mode 2 (BPSK 3/4 rate), mode 4 (QPSK 3/4 rate) and mode 5 (16QAM 9/16 rate) have almost the same E_b/N_o requirements, but their C/N performances differ due to the number of bits used during the modulation process. The performance degradation is due to the fact that the punctured convolutional code can not cope with large and deep fades in the frequency domain [26, 27, 44]. We can draw a simple operating line for the mode hierarchy using BER/PER performance against C/N:

$$Mode\ 1 > Mode\ 3 > Mode\ 2 > Mode\ 4 > Mode\ 5 > Mode\ 6 > Mode\ 7$$

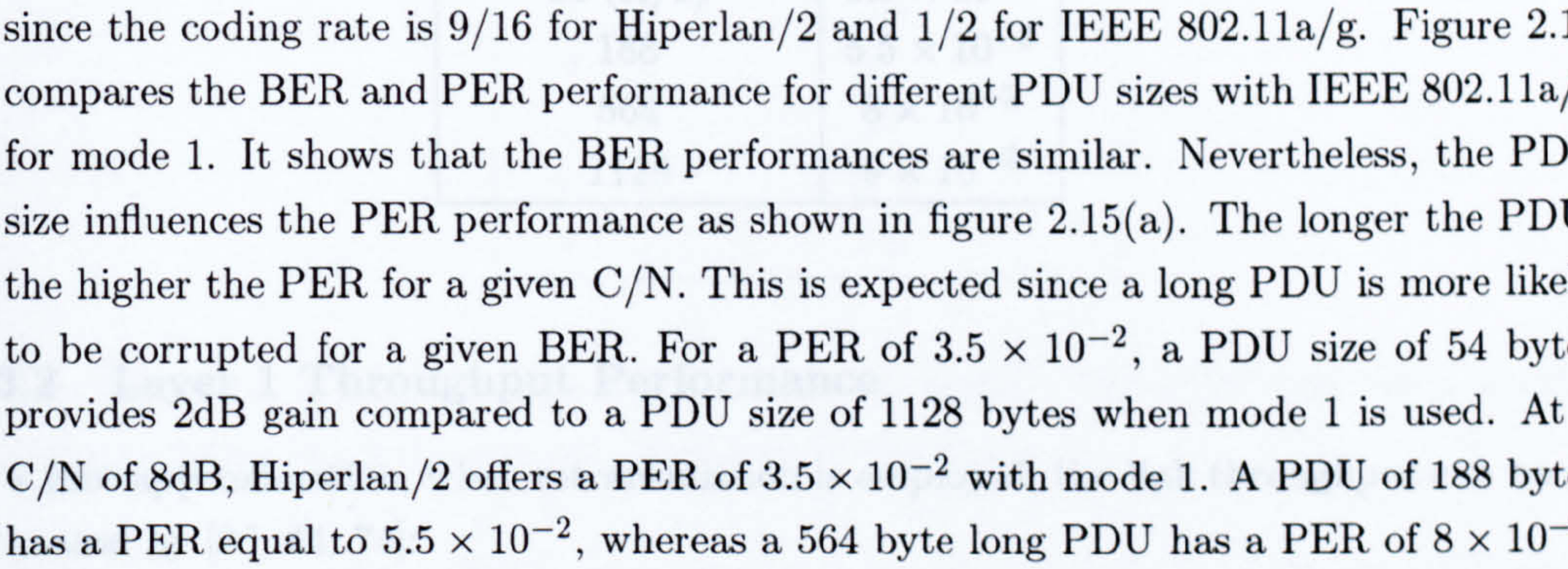
As outlined in section 2.2.3, the IEEE802.11a/g standard uses variable PDU lengths whereas Hiperlan/2 uses fixed PDU sizes. Due to their common PHY layer, the BER



(a) PER (b) BER

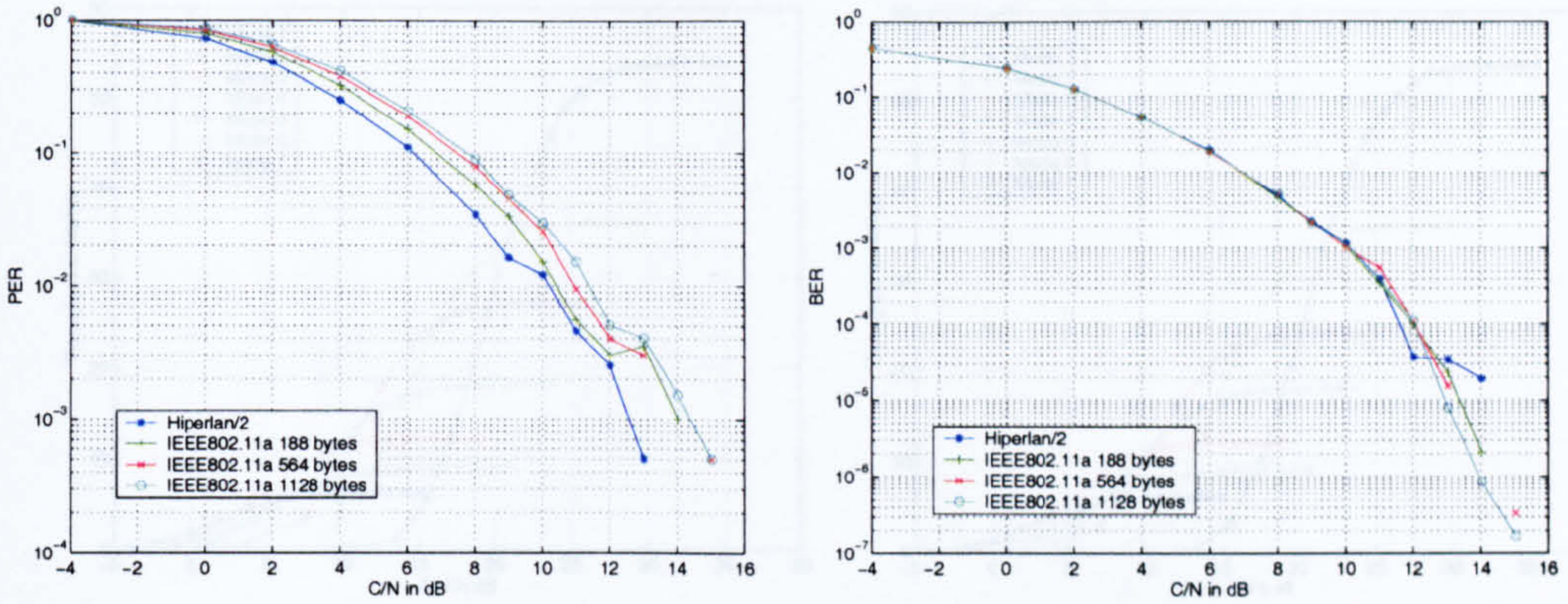
Figure 2.13: BER and PER Performance (PHY layer) for Hiperlan/2 versus C/N

performance is similar for the three standards [26, 27], regardless of the size of the packet to be transmitted. The PER performances are however affected. Figure 2.14 shows the BER and PER performance of IEEE 802.11a/g with a PDU size of 188 bytes versus C/N. Note that mode 5 presents different BER performance from Hiperlan/2 since the coding rate is 9/16 for Hiperlan/2 and 1/2 for IEEE 802.11a/g. Figure 2.15 compares the BER and PER performance for different PDU sizes with IEEE 802.11a/g for mode 1. It shows that the BER performances are similar. Nevertheless, the PDU size influences the PER performance as shown in figure 2.15(a). The longer the PDU, the higher the PER for a given C/N. This is expected since a long PDU is more likely to be corrupted for a given BER. For a PER of 3.5×10^{-2}, a PDU size of 54 bytes provides 2dB gain compared to a PDU size of 1128 bytes when mode 1 is used. At a C/N of 8dB, Hiperlan/2 offers a PER of 3.5×10^{-2} with mode 1. A PDU of 188 bytes has a PER equal to 5.5×10^{-2}, whereas a 564 byte long PDU has a PER of 8×10^{-2}. This is summarised in table 2.4.



(a) PER (b) BER

Figure 2.14: BER and PER Performance (PHY layer) for IEEE802.11a/g, PDU size = 188 bytes



(a) PER (b) BER

Figure 2.15: Performance (PHY layer) comparison for IEEE 802.11a/g, different packet length, Mode 1

Table 2.4: PER Performance (PHY layer) at C/N=8dB, different PDU sizes, Mode 1

PDU size in bytes	*PER*
54 (H/2)	3.5×10^{-2}
188	5.5×10^{-2}
564	8×10^{-2}
1128	9×10^{-2}

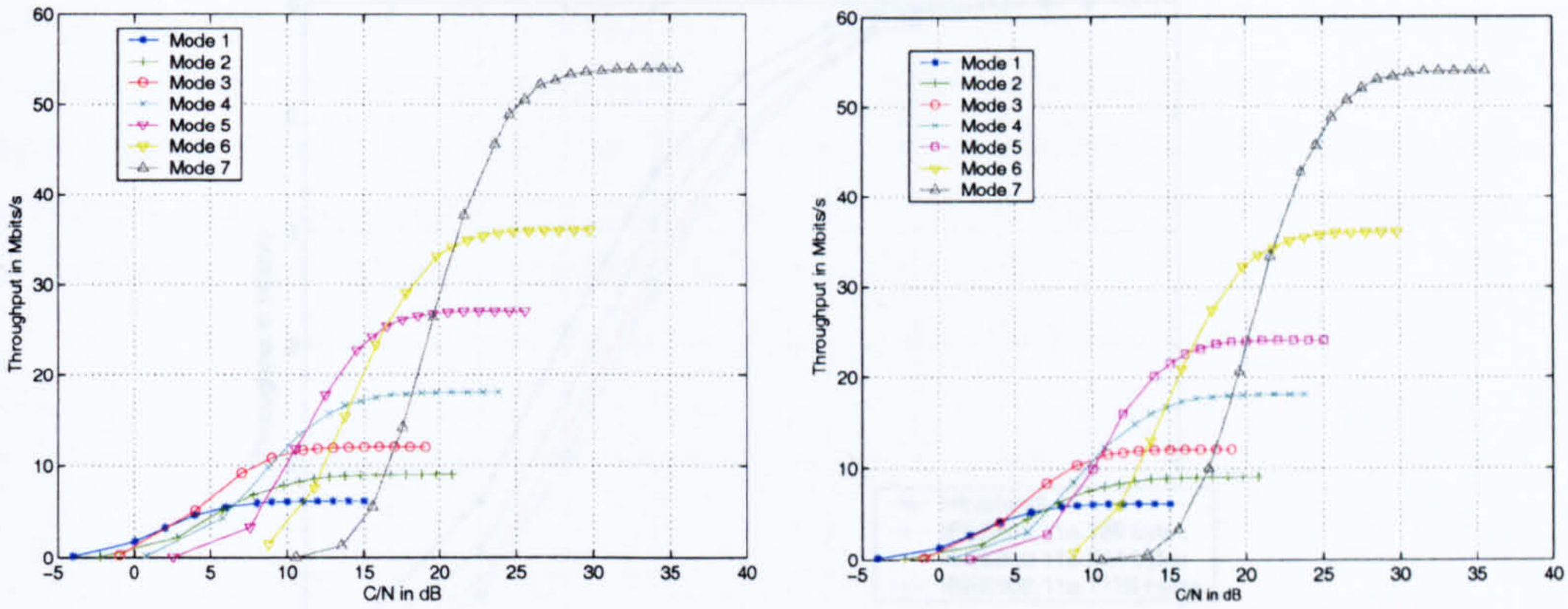
2.3.2 Layer 1 Throughput Performance

At a first approximation, when retransmission is employed, the link throughput can be estimated by [26, 44, 74]:

$$Throughput = R \times (1 - PER) \tag{2.6}$$

where R and PER are the nominal bit rate and the packet error rate of the link respectively for a specific operating mode. Nominal bit rates for the different modes are available in table 2.1. Figure 2.16 shows the throughput for the different modes versus C/N for Hiperlan/2 and IEEE 802.11a/g with a PDU size of 188 bytes. The mode offering the best throughput depends on the C/N. Table 2.5 gives an example of the different throughputs available at a C/N of 15dB. This provides the link with the possibility to choose the mode with the highest throughput. Given the various modes in which IEEE802.11a/g and Hiperlan/2 can operate, throughput provides a possible metric for link adaptation as detailed in chapter 7.

From equation 2.6, the throughput depends on the PER. Given that for a specific mode, a longer packet has a higher probability of error with IEEE 802.11a/g, the link throughput is lower for longer packet as shown in figure 2.17. Table 2.6 shows the different throughputs available for mode 1 with a C/N of 4dB for different PDU sizes.



(a) Hiperlan/2

(b) IEEE 802.11a/g - 188 bytes

Figure 2.16: Throughput Performance (PHY layer) for Hiperlan/2 and IEEE 802.11a/g

Table 2.5: Throughput (PHY layer) available for Hiperlan/2 at C/N=15dB

Mode	*Throughput in Mbits/s*
1	5.9
2	8.8
3	11.9
4	16
5	20
6	11.7
7	0.5

The difference in link throughputs are however small. The PDU size has a greater impact on the MAC throughput, as shown in chapter 3.

Table 2.6: Throughput (PHY layer) available for IEEE802.11a/g, different PDU sizes, C/N=4dB, mode 1

PDU size in bytes	*Throughput in Mbits/s*
54 (H/2)	4.55
188	4.05
564	3.75
1128	3.5

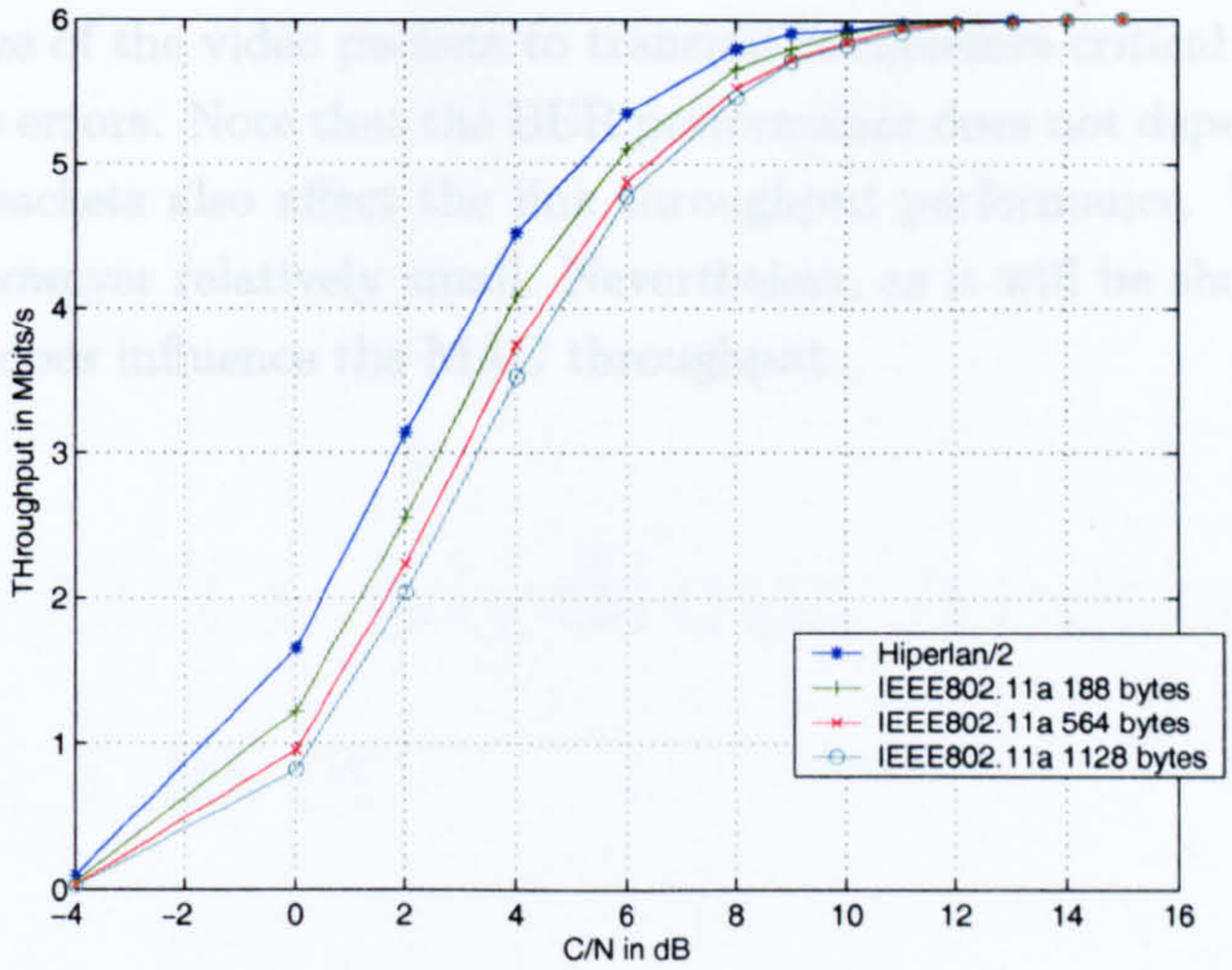


Figure 2.17: Throughput Performances comparison, different PDU size, Mode 1

2.4 Conclusion

This chapter presented an overview of COFDM-based WLANs, with focus given to the PHY layers of the Hiperlan/2 and IEEE802.11a/g WLANs standards at 2.4 and 5.2GHz. A detailed description of OFDM has been given. Performance results have been presented in terms of BER, PER and throughput for all the operating modes available. Results were obtained using a simulator developed previously within the University of Bristol.

From the performance results, attention is drawn to the fact that the BPSK 1/2 rate mode (mode 1) provides the most reliable transmission mode, at the expense of the lowest nominal bit rate (6 Mbits/s). On the other hand, 64 QAM 3/4 rate (mode 7) offers unreliable transmission, but with the highest nominal bit rate (54 Mbits/s). The other operating modes stand in between these two extremes. The choice of the transmission mode is therefore critical. A trade-off has to be found between robust transmission with a lowered transmission rate and unreliable transmission with a high transmission rate. Because of the channel errors, the nominal throughput of each mode is however reduced when retransmission is used. The available bit rate depends therefore on the PER at the PHY. A real-time multimedia transmission may require very high bit rates, especially for broadcast video. The operating mode should therefore be careful chosen. Note that, given that various overheads on the protocol stack reduce the available bit rate at the application layer, as detailed in chapters 3 and 6, a link with a maximum throughput of 6 Mbits/s at the PHY layer may not be sufficient to support such applications.

Since the PHY layer is common to both standards, BER performances are similar. However, they differ in their structure with IEEE802.11a/g allowing variable length PDUs (up to 4096 bytes), whereas Hiperlan/2 only transmits fixed size PDUs (54 bytes). The PDU size has a significant impact on the PER performance. Large video

packets, leading to large PHY PDUs, will be more likely corrupted than smaller video packets. The size of the video packets to transmit is therefore critical for the resilience of the system to errors. Note that the BER performance does not depend on the packet length. Large packets also affect the link throughput performance. The difference in throughput is however relatively small. Nevertheless, as it will be shown in chapter 3, the packet size does influence the MAC throughput.

Chapter 3

IEEE 802.11 Medium Access Control Layer

In this chapter, a detailed overview the IEEE 802.11 Medium Access Control (MAC) layer is given. The Hiperlan/2 MAC is developed in Appendix B. The chapter is organised as follows. After an introduction in section 3.1, section 3.2 details the mandatory Distributed Coordination Function (DCF) and discusses the impacts for video transmission. The optional Point Coordination Function (PCF) mode is discussed in section 3.3. In section 3.4, a study based on Markov chains, based on [25] highlights the impact of the number of users on the DCF performance. Section 3.5 presents the QoS limitations of the IEEE 802.11 MAC, leading to conclusions in section 3.6.

3.1 Introduction

Current wireless LANs at 2.4 and 5.2GHz, as discussed in chapter 2, support bit rates up to 54 Mbits/s. This throughput is however calculated at the PHY layer, this does not take into account any overheads from the upper layers. Real-time Multimedia and/or video transmission requires high bit rates at the application layer and are time bounded. Underlying network layers should be able to support these rates and with minimum delay and jitter. In a wireless LAN where the channel resources are shared, the way stations access the medium is critical for the system performance especially in terms of throughput and delay. Real-time video transmissions, and other home entertainment streams, are very sensitive, and throughput and delay constraints should be guaranteed by underlying networks. Here, the IEEE 802.11 MAC Medium Access Control (MAC) layer [39], which is responsible for sharing and providing fair access to the channel, is studied.

The IEEE 802.11 MAC lies on the top of the IEEE 802.11a/b/g PHY layer. It provides shared asynchronous access to the wireless channel and offers two operating modes. The first one is mandatory and is called the Distributed Coordination Function (DCF). It defines distributed access for an ad-hoc network. The second operating mode is called the Point Coordination Function (PCF), and is optional. It defines centralised access for an infrastructure network [39]. Because PCF is thought to be too complex

[75], PCF has only been implemented in a few chipsets and a short study is presented here. Both modes are based on the same Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) access protocol. The DCF is used as a basis for the PCF and is also defined as the Contention Mode, whereas the PCF is known as the Contention Free Mode. If the optional PCF is used, it shall then alternate with the DCF in a super-frame with a Contention Period (CP under DCF) and a Contention Free Period (CFP under PCF) as shown in figure 3.1. Because it is the main access scheme implemented on real cards, the focus of the analysis will given to the DCF.

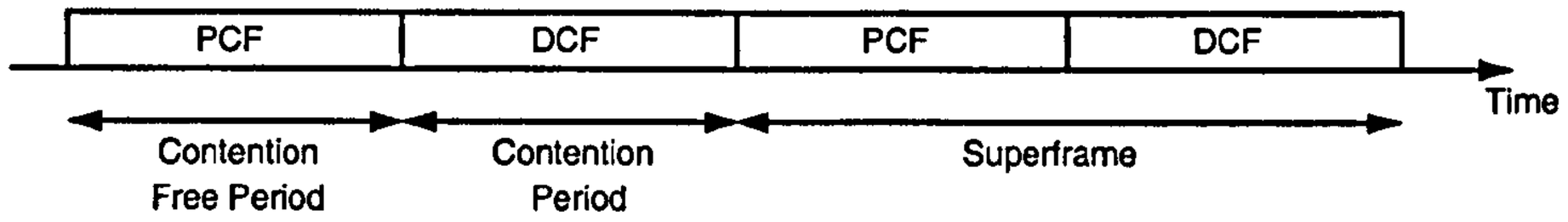


Figure 3.1: Contention and Contention Free Period in a super-frame

The transmitted entity is called a IEEE 802.11 MAC frame and its general data format is described in figure 3.2 [39, 76, 77]. The MAC header is 30 byte long and is composed of the Frame Control (FC) field (containing information on the protocol versions, power management, retry), a Duration/ID field, four addresses (receiver, transmitter, source and destination) and a Sequence Control (SC) field (containing the sequence number and the fragment number). The Frame Body field contains the payload and has a maximum length of 2312 bytes for IEEE802.11b/g and 4096 bytes for IEEE802.11a. Appended at the tail of the body, the Frame Check Sum (FCS) field contains a 32-bit Cyclic Redundancy Check (CRC) code calculated over all the fields of the MAC header and Frame Body field [39, 76]. The FCS is used at the receiver to check whether the received packet is correct or not.

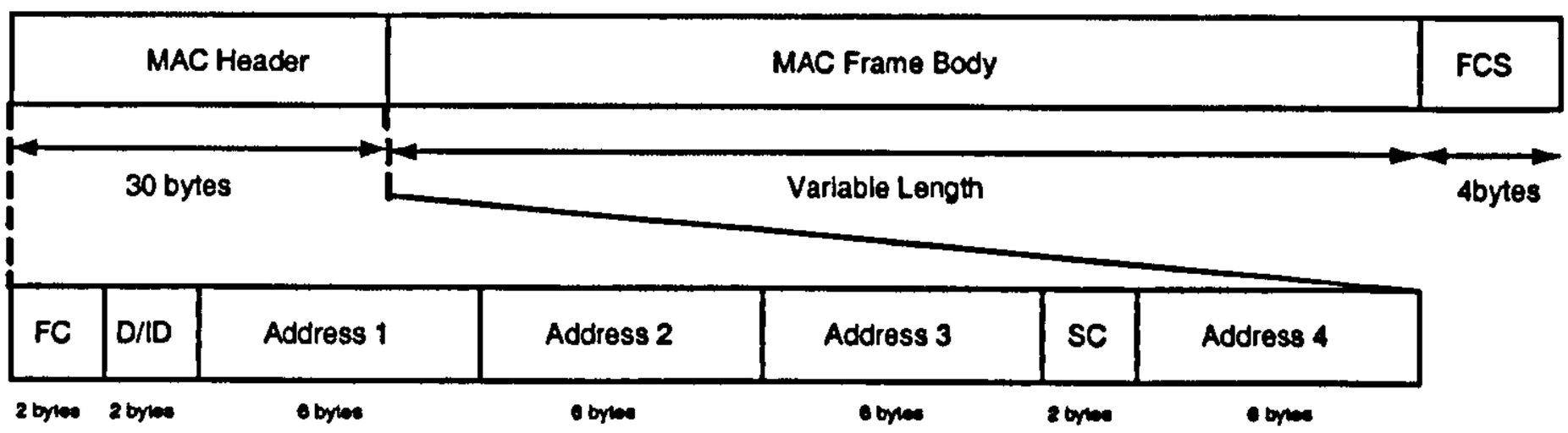


Figure 3.2: General format of a MAC Frame

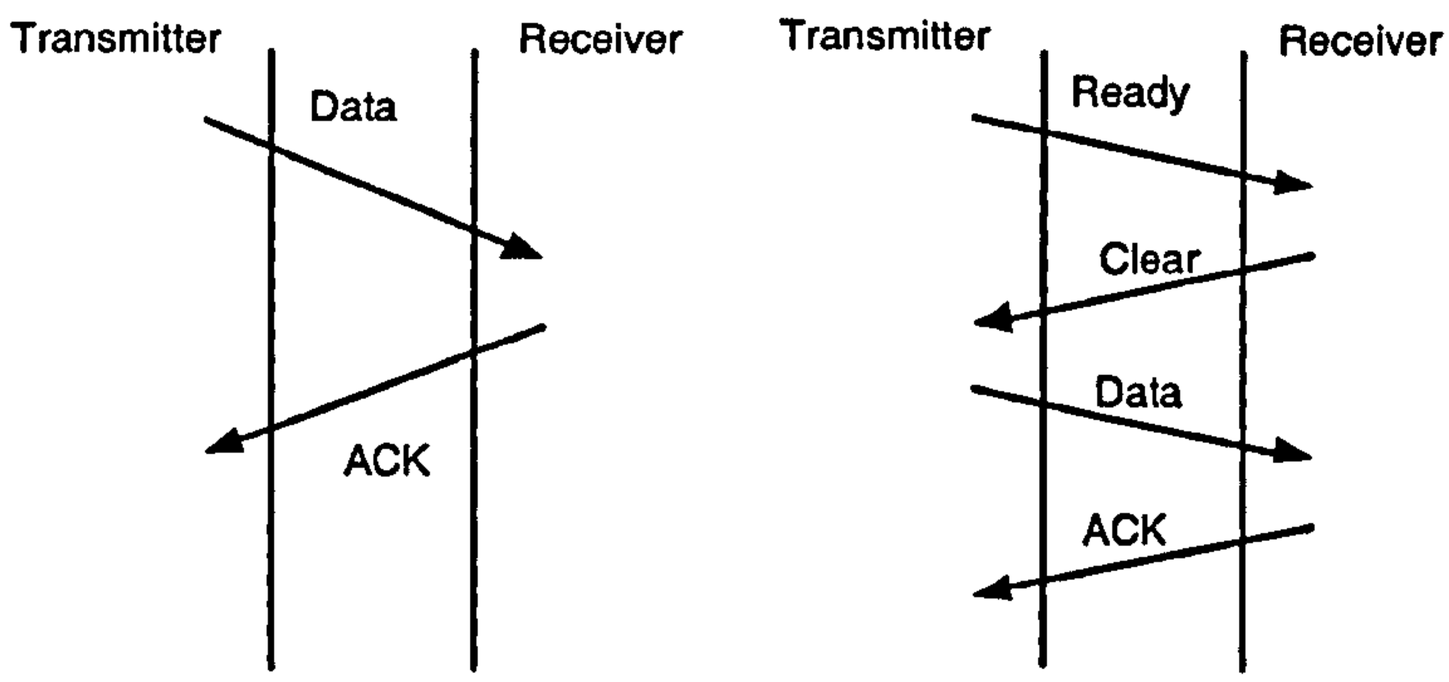
3.2 Distributed Coordination Function

3.2.1 Access Mechanisms

The DCF provides basic asynchronous and contention-based shared access to the medium. It is a distributed scheme for ad-hoc networks and is based on CSMA/CA. CSMA/CA

is also known as 'Listen before Talk'; that is, every user before attempting any transmission listens whether to hear somebody else is already using the channel [78]. Before a station starts a transmission, it shall sense the wireless medium to determine if another station is transmitting. If the medium is sensed to be idle, the transmission may proceed. If the medium is sensed to be busy, the station shall defer until the end of the current transmission to avoid creating collision [22, 24]. An exhaustive performance analysis of carrier sense protocols is available in [78] and [79].

The DCF defines two techniques for transmission. The basic scheme is mandatory and is known as a two-way handshaking technique, see figure 3.3(a). It is characterised by the immediate transmission of a positive acknowledgment (ACK) by the receiver if the current packet has been successfully received. If the ACK is not received by the source after a timeout, it is assumed that either a collision has occurred or the transmission has not been successful and the data transmission is rescheduled. The second scheme is optional and is known as a four-way handshaking technique, see figure 3.3(b). The transmitting station uses a Ready to Send (RTS) notice to inform the receiver and to reserve the channel. The receiver shall then reply by acknowledging with a Clear to Send (CTS). After the reception of the CTS, the transmission shall proceed. As in the basic scheme, the receiver shall also immediately acknowledge the transmitted data packet if successfully received. If the CTS is not received by the source after a timeout, it is assumed that either a collision has occurred or the transmission has not been successful and the RTS transmission is rescheduled [80]. With this scheme, collisions may only occur on the first RTS frame. RTS/CTS is used to combat the 'hidden node problem' which, occurs when a station is causing interference due to not being able to detect another node's transmissions. RTS/CTS also enhances system performance by reducing the duration of collisions when long packets are transmitted [25], as shown in section 3.4. For both schemes, at the receiver, the FCS determines whether the received packet is corrupted or not.

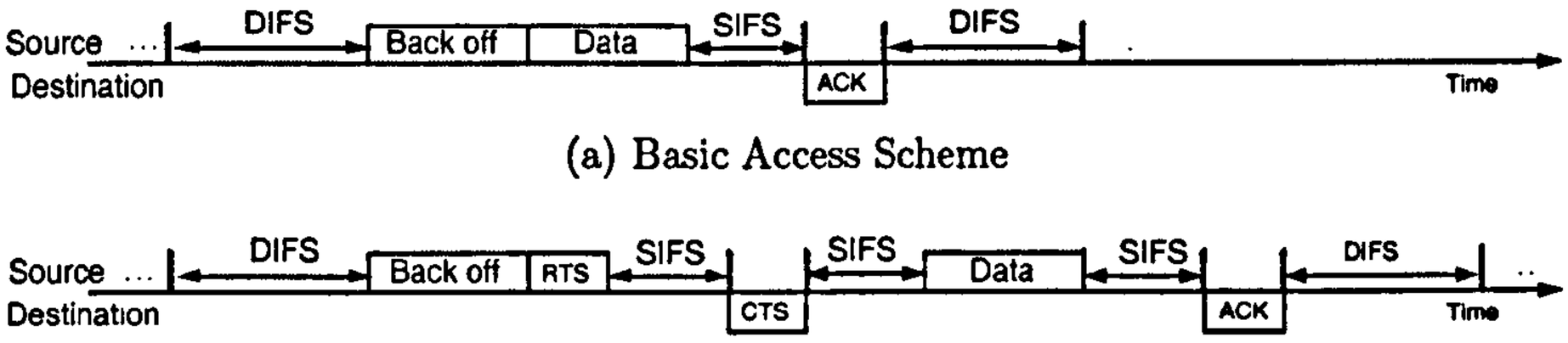


(a) Two-way handshaking (b) Four-way handshaking

Figure 3.3: Handshaking Mechanism

The IEEE 802.11 MAC uses Inter Frame Space (IFS) timing to control the access to the channel. Each station is allowed to transmit only if it has sensed the medium

to be idle for at least a Distributed IFS (DIFS). In addition, it shall also wait for a random back-off after DIFS, prior to attempting to transmit. The time duration between the reception of data and the transmission of an acknowledgment is called the Short IFS (SIFS) [39]. Figure 3.4(a) shows the cycle of the basic access mechanism for a successful transmission. After this cycle, all the stations may contend again for access to the medium. Figure 3.4(b) shows the cycle of the RTS/CTS access mechanism for a successful transmission [37, 80, 81].



(a) Basic Access Scheme

(b) RTS/CTS Access Scheme

Figure 3.4: DCF Access Mechanisms

The Contention for the medium is similar for both schemes. RTS and CTS frames are introduced at SIFS intervals. However, SIFS is always smaller than DIFS in order to prevent any other station trying to access the medium. Before attempting any transmission, each station shall wait at least a DIFS interval for the medium to be idle. Thus, by setting SIFS to be smaller than DIFS, priority is given to the current transmitting station. Collisions are unable to occur as a result of the inability of another station to detect the medium as being idle for a DIFS until the end of the ACK. Moreover, RTS and CTS packets contain a duration field indicating the amount of time the channel is reserved [82]. This information is collected by others stations listening to the channel. They all update their *Network Allocation Vector* (NAV) indicating the period of time that remains before the channel will become idle, see figure 3.5. The NAV is referred to as virtual carrier sensing [83], and even if a station is hidden from either the transmitting or the receiving station, it can suitably defer further transmission and avoid collision [84].

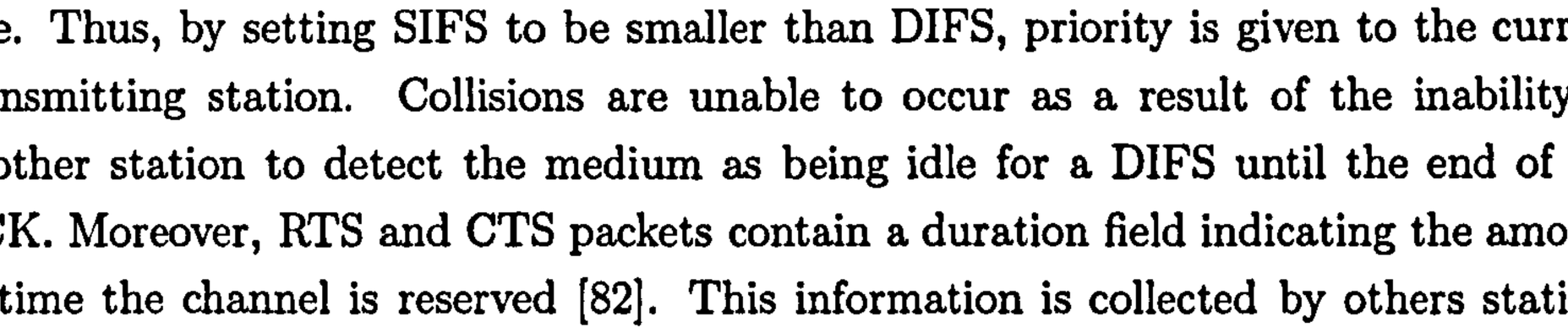


Figure 3.5: Network Allocation Vector Mechanism

3.2.2 Back-off Procedure

The back-off time following the DIFS is slotted and a station is allowed to transmit only at the beginning of each slot. This slot time size is the time needed for any station to detect the transmission of a packet from another station [25]. DCF uses an exponential back-off scheme to determine the random back-off timing. The back-off time is determined by:

$$Backoff\ Time = Backoff\ Counter \times Slot\ Time \tag{3.1}$$

where the back-off counter is uniformly and randomly chosen in the range [0, CW]. CW is the Contention Window. The back-off counter is decremented when the medium is sensed to be idle and then frozen if the medium is sensed to be busy. The counter is resumed when the medium is sensed to be idle again after DIFS. Transmission may proceed when the counter has reached zero. Figure 3.6 shows an example of the contention window process with basic access. After sensing the medium as idle for a DIFS time, Tx A and Tx B randomly set their back-off counters to 4 and 7 respectively. The back-off (BO) counters are decremented until either BO(A) or BO(B) become zero. BO(A) reaches zero when BO(B) is equal to 3, and consequently Tx A starts transmitting data whilst Tx B freezes its counter. Once the transmission cycle is finished, Tx A and Tx B restart sensing the medium. Tx B shall reactivate its BO(B), which was frozen at 3, whereas Tx A uses a new one (BO(A) = 5).

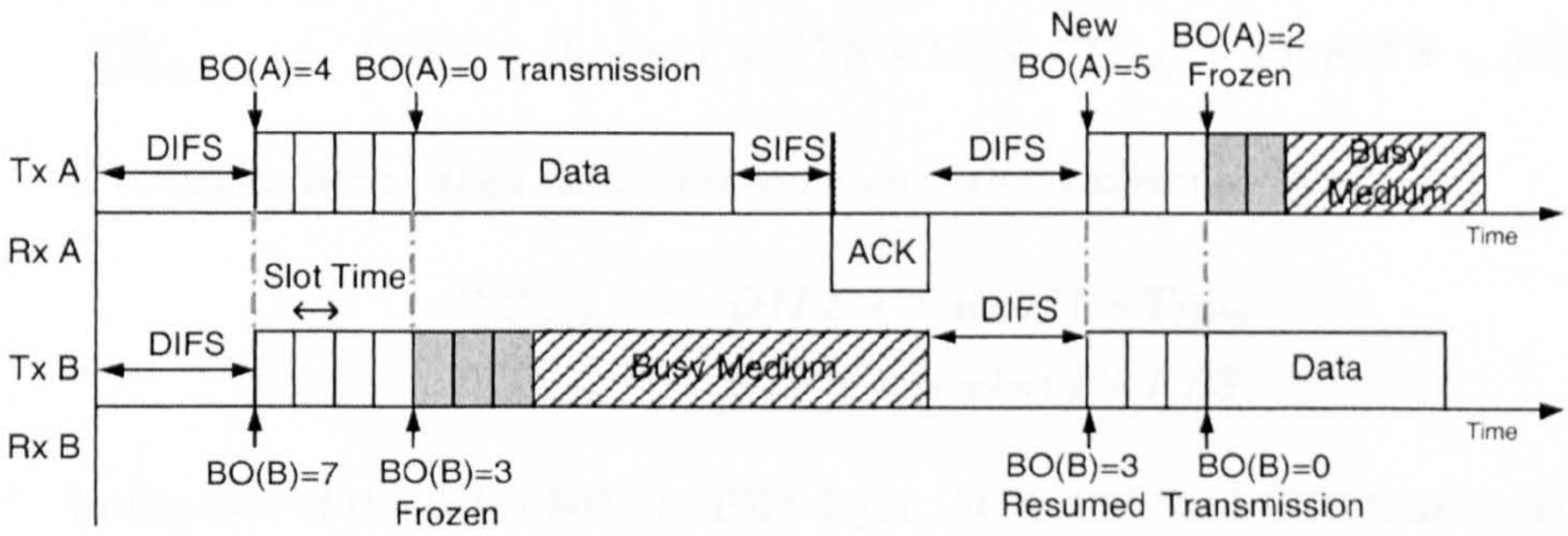


Figure 3.6: Contention Window Mechanism with the basic access

The IEEE 802.11 MAC implements mandatory Stop and Wait retransmission (ARQ). The contention window CW depends on the number of failed transmissions of the current frame. A transmission is considered as failed either when a collision has occurred, i.e. when two or more stations start transmissions simultaneously in the same slot time, or when no ACK or CTS frames are received after sending a data packet or an RTS frame respectively. Collisions occur when the back-off counters of two or more stations reach zero at the same time. No received ACK occurs when the data packet is corrupted, when the ACK collides with another packet, or when the ACK is corrupted. No received CTS occurs when the RTS frame is corrupted, when the CTS collides with another packet, or when the CTS frame is corrupted.

The CW is initially set to CW_{min} for the first transmission attempt. In order to reduce the probability of collision, after each failed transmission, the CW is doubled up to:

$$CW_{max} = 2^m \times (CW_{min} + 1) - 1 \tag{3.2}$$

where m is called the maximum back-off stage. Once it reaches CW_{max}, it remains at this value until it is reset. CW is reset to CW_{min} after a successful transmission. The value of CW is then:

$$CW = 2^i \times (CW_{min} + 1) - 1 \; if \; 0 \leq i \leq m \tag{3.3}$$
$$CW = CW_{max} \; if \; m \leq i \tag{3.4}$$

where i represents the number of unsuccessful attempts.

3.2.3 Throughput Derivation

From figures 3.4(a) and 3.4(b), it can be seen that successful cycle durations with the basic and RTS/CTS schemes are:

$$T_{success}^{basic} = DIFS + Backoff + T_{Data} + SIFS + ACK \tag{3.5}$$
$$T_{success}^{rts} = DIFS + Backoff + RTS + CTS + T_{Data} + 3 \times SIFS + ACK \tag{3.6}$$

If collisions occur, then these two equations are transformed into:

$$T_{collision}^{basic} = DIFS + Backoff + T_{Data} \tag{3.7}$$
$$T_{collision}^{rts} = DIFS + Backoff + RTS \tag{3.8}$$

In the case of the IEEE 802.11a PHY layer, ACK, RTS and CTS frames are transmitted with mode 1. T_{Data} is mode dependent as well as packet length dependent and is shown in equation 3.9 where $\lceil . \rceil$ corresponds to the ceiling function. The total throughput for both schemes is then given by equation 3.10.

$$T_{Data} = T_{PLCPpreamble} + T_{PLCPheader} + \left\lceil \frac{N_{MAC\ Header} + N_{Data} + N_{FCS}}{N_{DBPS}} \right\rceil \times T_s \tag{3.9}$$

$$Throughput = \frac{N_{Data}}{T_{success}} \tag{3.10}$$

$T_{PLCPpreamble}$ and $T_{PLCPheader}$ are the durations of the PLCP preamble and header respectively of the MAC frame. Their corresponding values are available in table 2.2 in section 2.2.3. $N_{MAC\ Header}$, N_{Data} and N_{FCS} are the number of bits in the MAC header, in the payload and in the FCS field respectively (see section 3.1). N_{DBPS} is

Table 3.1: MAC Timing Parameters

	IEEE802.11a	IEEE802.11b/g
Slot Time (μs)	9	20
SIFS (μs)	16	10
DIFS = SIFS+2Slots Time (μs)	34	50
PIFS = SIFS+Slot Time (μs)	25	30
CW_{min}	15	31
CW_{max}	1023	1023
ACK (μs)	44 (mode 1)	Mode dependent
Av. Backoff (μs)	67.5	310
m	6	5

the number of coded bits per OFDM symbol for the current operating mode, as defined in table 2.3 in section 2.2.2.3. T_s is the OFDM symbol duration. Apart from N_{Data} and N_{DPBS}, all the parameters are fixed, and the throughput depends on the packet length and the operating mode.

3.2.4 MAC Parameters

The IFS timings and the contention window parameters are PHY dependent, as shown in table 3.1. As stated in equation 3.11, the Point Coordination IFS (PIFS) is always smaller than The DIFS and larger than the SIFS and its use is explained in appendix 3.3.

$$SIFS < PIFS < DIFS \tag{3.11}$$

Manufacturers often implement IEEE 802.11g backward compatible with 802.11b, so companies can easily upgrade their networks. In this case, IEEE 802.11b and g share similar MAC parameters. If IEEE 802.11g is not backward compatible, then 802.11a parameters are used. For IEEE802.11a, the maximum back-off stage m is reached after 6 failed transmissions, whereas for IEEE802.11b/g, it is reached after only 5 failed transmissions due to a larger CW_{min}.

The average back-off defines the back-off duration for 'lightly loaded networks', i.e. when each station has access to the channel after the first back-off attempt [27, 80]. Upon this assumption, the contention window length is therefore always CW_{min}. The average or expected back-off duration is then:

$$Average\ Backoff = Slot\ Time \times \frac{CW_{min}}{2} \tag{3.12}$$

3.2.5 Fragmentation

Under the DCF, the MAC may fragment large packets from the upper layers into smaller MAC frames. This allows for corrupted fragments to be retransmitted, rather

than the whole frame [39]. If one fragment of a long packet is lost (corruption or collision), only the missing fragment needs to be retransmitted. Moreoever, fragmentation improves the reliability by increasing the probability of successful fragment transmission with smaller packets being less likely to be corrupted. Each fragment is sent as an independent transmission and is acknowledged separately [24]. However, fragments are sent in bursts. Once the station has contented for the channel, it can transmit fragments with a SIFS interval between the ACK and the next fragment. Whenever one ACK is not received, the station has to content again for the channel with the back-off procedure. The station shall resume the transmission from the last missing packet. All the fragments should have the same length except for the last one. Fragmentation is applied whenever the packet length exceeds a certain threshold. The fragmentation threshold takes its value between 256 and 2047 bytes [39] but is not standardised and left to manufacturers. Fragments are reassembled at the receiver, using information contained in the MAC header.

3.2.6 DCF Performance

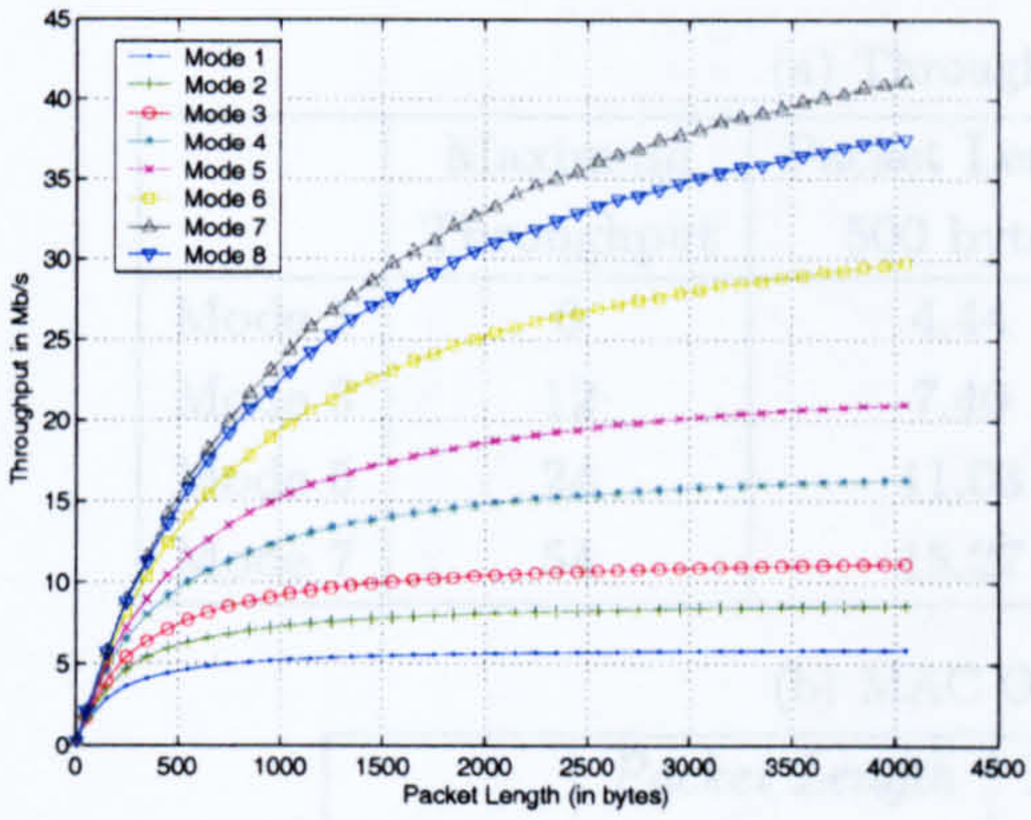
The simulations performed and the results presented in this section assume a 'lightly loaded network'. Impacts on video requirements in terms of throughput and delay are also discussed. Unless specified, the IEEE 802.11a PHY is used with the IEEE 802.11a MAC parameters.

3.2.6.1 Throughput

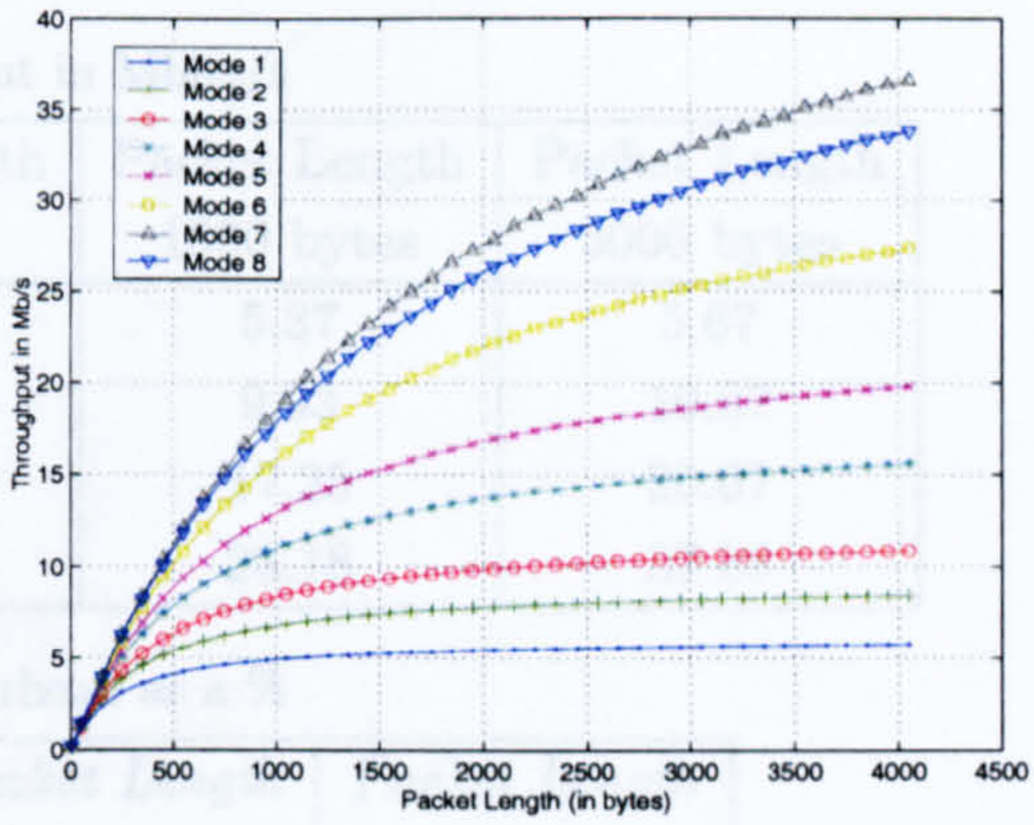
Figure 3.7 shows the simulated throughput performance of the system for the 8 operating modes for different packet lengths and for the basic DCF access and the RTS/CTS access scheme [80]. As previously mentioned, it can be seen that the throughput is packet length dependent. This is especially true for higher modes. This can be explained by the fact that the MAC overheads are considerably longer compared with the data transmission duration if packets are small. This is shown in figure 3.8. Applications requiring high bit rates are therefore likely to use larger packets. Table 3.2 summarises the throughput efficiency and overhead efficiency for different packet lengths (500, 1500 and 3000 byte long packets) and for modes 1, 3, 5 and 7. Mode 5 with 500 byte long packets offers a throughput of 11 Mbits/s, whereas 1500 byte packets leads to a throughput of 17.25Mbits/s. However, longer packets are more likely to be corrupted (see section 2.3.1).

Figure 3.9 compares the throughput and MAC overhead efficiency of the basic and the RTS/CTS access schemes. It can be seen that the basic scheme offers better throughput than the RTS/CTS access technique. The RTS/CTS scheme decreases the efficiency since it transmits two additional frames without payload and two SIFS are introduced. This is only true under the assumption that each station has access to the medium after the first attempt, i.e. in a 'lightly loaded network' [80]. This is not valid when the number of users increases. This case will be detailed in section 3.4.

Because of the MAC overhead, the throughput efficiency of the PHY as depicted



(a) Basic Access



(b) RTS/CTS Access

Figure 3.7: Throughput - DCF

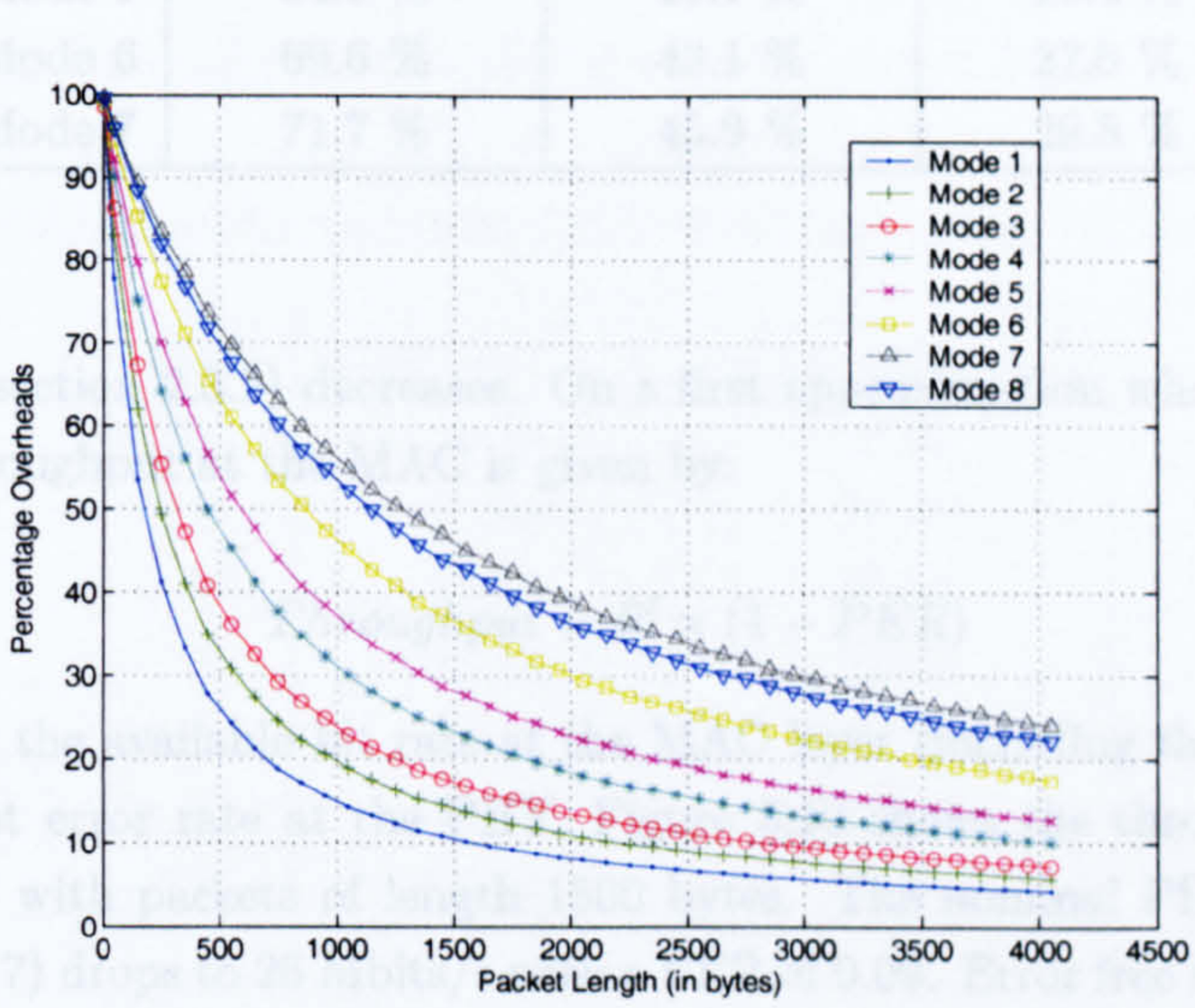
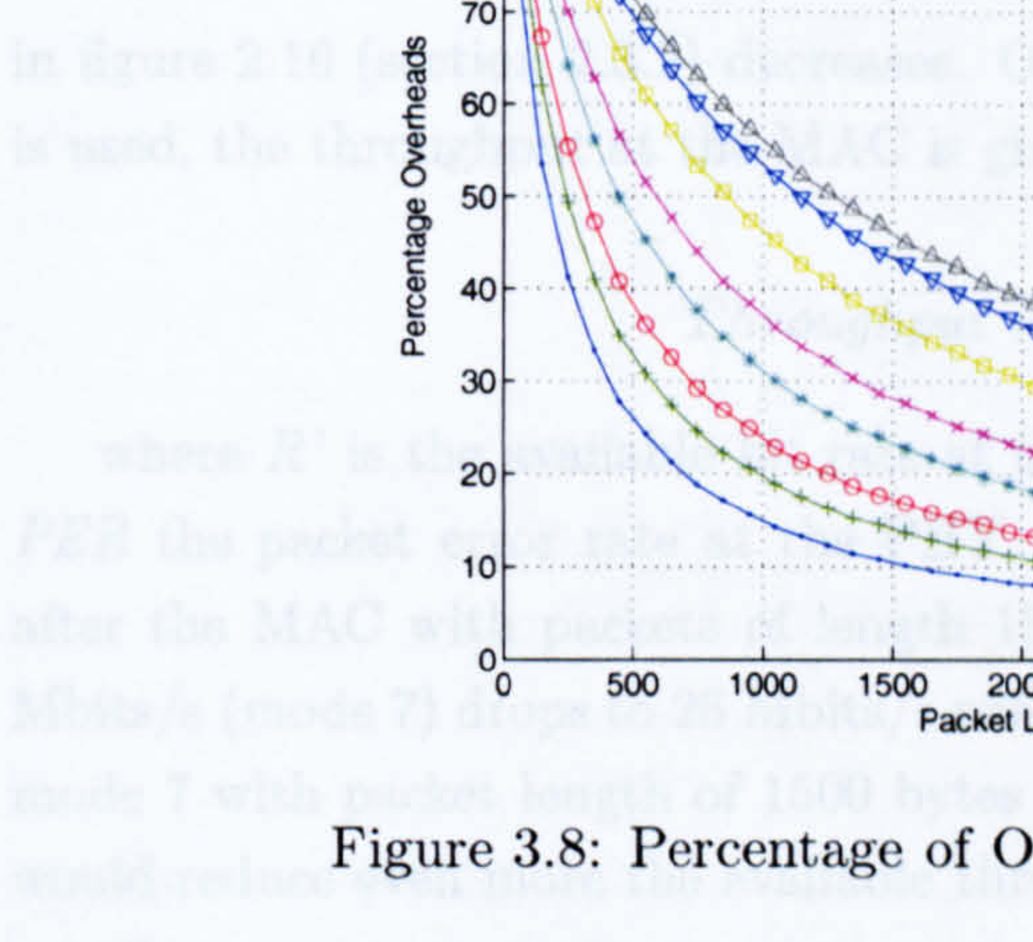
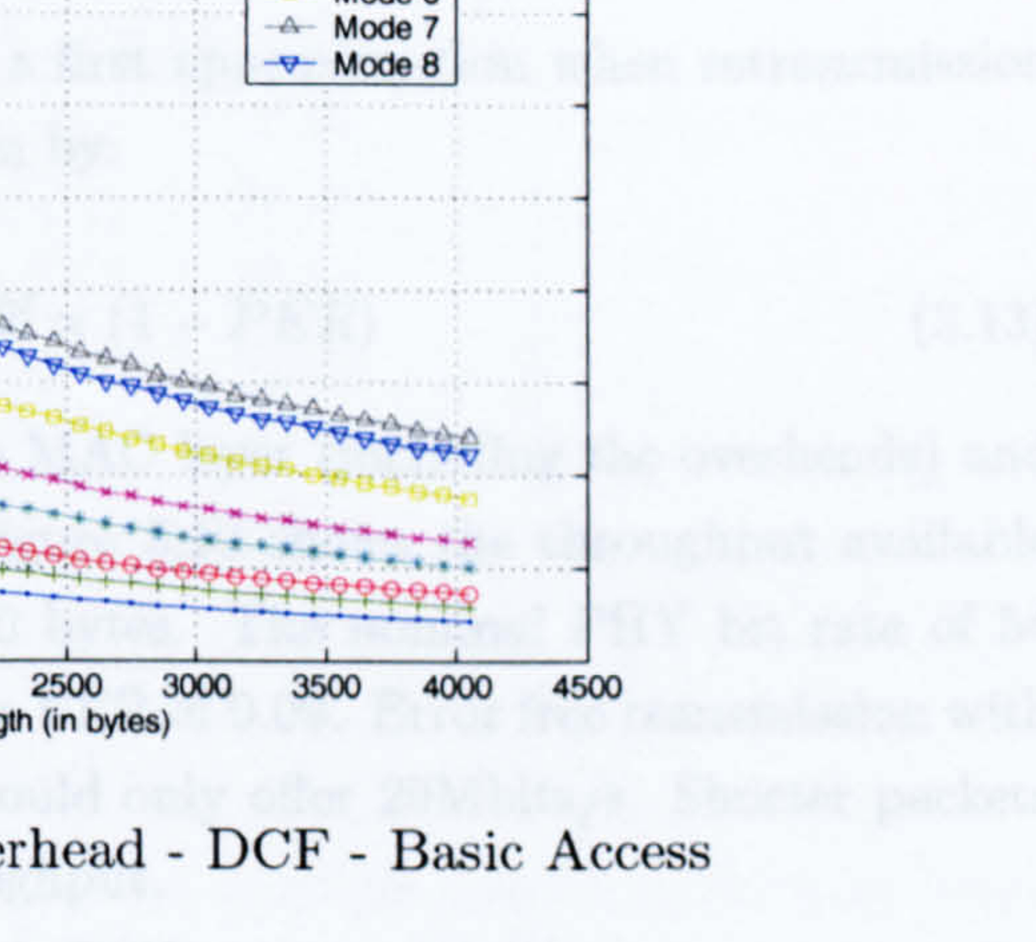


Figure 3.8: Percentage of Overhead - DCF - Basic Access



(a) Throughput Comparison



(b) MAC overhead Comparison

Figure 3.9: Basic Access - RTS/CTS Access Comparison for Mode 3

Table 3.2: MAC Throughput Efficiency - DCF - Basic Access

(a) Throughput in Mbits/s

	Maximum Throughput	Packet Length 500 bytes	Packet Length 1500 bytes	Packet Length 3000 bytes
Mode 1	6	4.44	5.37	5.67
Mode 3	12	7.40	9.93	10.87
Mode 5	24	11.03	17.25	20.07
Mode 7	54	15.27	28.18	37.88

(b) MAC Overhead as a %

	Packet Length 500 bytes	*Packet Length 1500 bytes*	*Packet Length 3000 bytes*
Mode 1	25.8 %	10.1 %	5.5 %
Mode 3	38.4 %	17.2 %	9.4 %
Mode 5	54.0 %	28.1 %	16.4 %
Mode 6	69.6 %	43.1 %	27.5 %
Mode 7	71.7 %	45.9 %	29.8 %

in figure 2.16 (section 2.3.2) decreases. On a first approximation when retransmission is used, the throughput at the MAC is given by:

$$Throughput = R' \times (1 - PER) \tag{3.13}$$

where R' is the available bit rate at the MAC layer (including the overheads) and PER the packet error rate at the PHY. Figure 3.10 shows the throughput available after the MAC with packets of length 1500 bytes. The nominal PHY bit rate of 54 Mbits/s (mode 7) drops to 26 Mbits/s with a PER of 0.09. Error free transmission with mode 7 with packet length of 1500 bytes would only offer 29Mbits/s. Shorter packets would reduce even more the available throughput.

These previous results were obtained with IEEE 802.11a MAC parameters detailed in table 3.1. However, with IEEE 802.11g backward compatible with b, the MAC parameters change and so does the performance. Table 3.3 shows the difference in overheads for the two MAC parameters for the DCF case with basic access, for a packet size of 1500 bytes. It can be seen that due to larger DIFS, Slot time and Contention window, overheads with IEEE 802.11 b/g are greater than for IEEE 802.11a. Even with a similar PHY layer, 802.11a and g cannot provide similar throughputs if 802.11g is chosen to be backward compatible with b.

As described in table 1.1 in chapter 1, bit rates up to 20Mbits/s may be required for video, especially for HDTV. The throughput analysis of the basic access of the DCF shows that the packet length at the MAC layer has a very strong influence on the throughput performance. Small video packets coming from the upper layers will lead to very large overheads and therefore to reduce throughput. Inversely, for a video

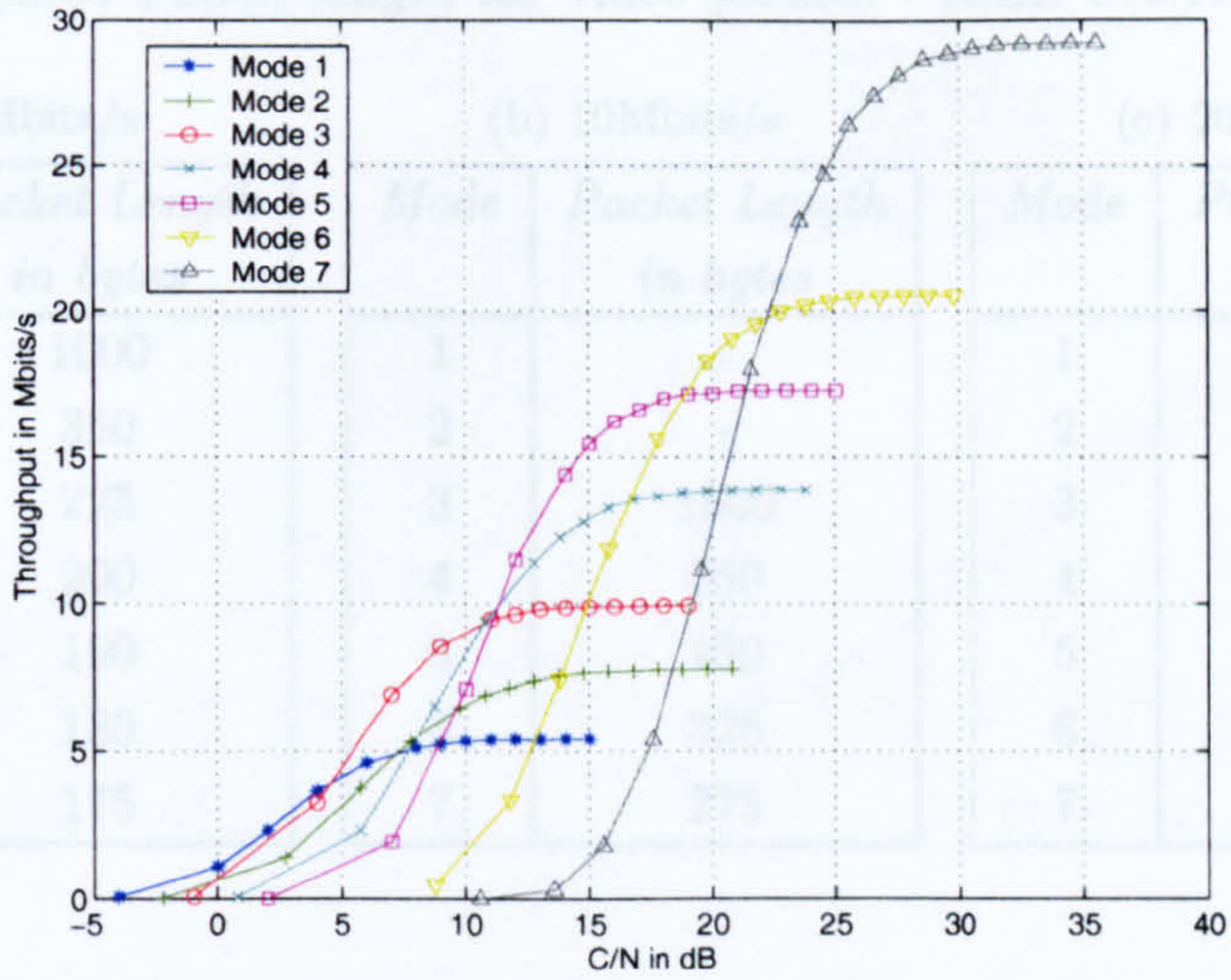


Figure 3.10: Throughput after the MAC with packet length of 1500 bytes

Table 3.3: MAC Overhead Comparison IEEE 802.11a/b/g (in %)- DCF - Basic Access - 1500 bytes

Mode	*IEEE 802.11a*	*IEEE 802.11b/g*
1	10.15	18.97
2	13.92	25.36
3	17.25	30.76
4	23.16	39.63
5	28.12	46.36
6	43.13	56.26
7	45.96	65.71

encoded at a specific rate to be transmitted with a single user, a minimum length for video packets is required, as shown in table 3.4 for video bit rates of 5, 10 and 20 Mbits/s transmitted over IEEE 802.11a. If IEEE 802.11b/g is used, overheads are larger and therefore constraints on the video packet lengths are stronger.

3.2.6.2 Influence of ARQ

As mentioned in section 3.2.2, the back-off procedure in the DCF has been designed so that it minimises collisions on the channel. This procedure is also applied whenever a transmission fails, meaning that if the channel conditions are bad, corrupted packets are retransmitted with a doubling of the CW, until a maximum value is reached. The number of ARQs allowed is left to manufacturers. These ARQs introduce delays in the DCF as shown in figure 3.11 for the IEEE 802.11a and IEEE 802.11b/g MAC parameters. It was assumed that no collisions occurred and that data are transmitted with mode 1 (6Mbits/s). In the case of RTS/CTS access, the delay is independent of the packet length since only the loss of RTS packet indicates a failed transmission.

Table 3.4: Required Packet length for video packets - IEEE 802.11a - DCF - Basic Access

(a) 5Mbits/s

Mode	*Packet Length in bytes*
1	1000
2	350
3	225
4	200
5	190
6	180
7	175

(b) 10Mbits/s

Mode	*Packet Length in bytes*
1	-
2	-
3	1500
4	550
5	450
6	325
7	275

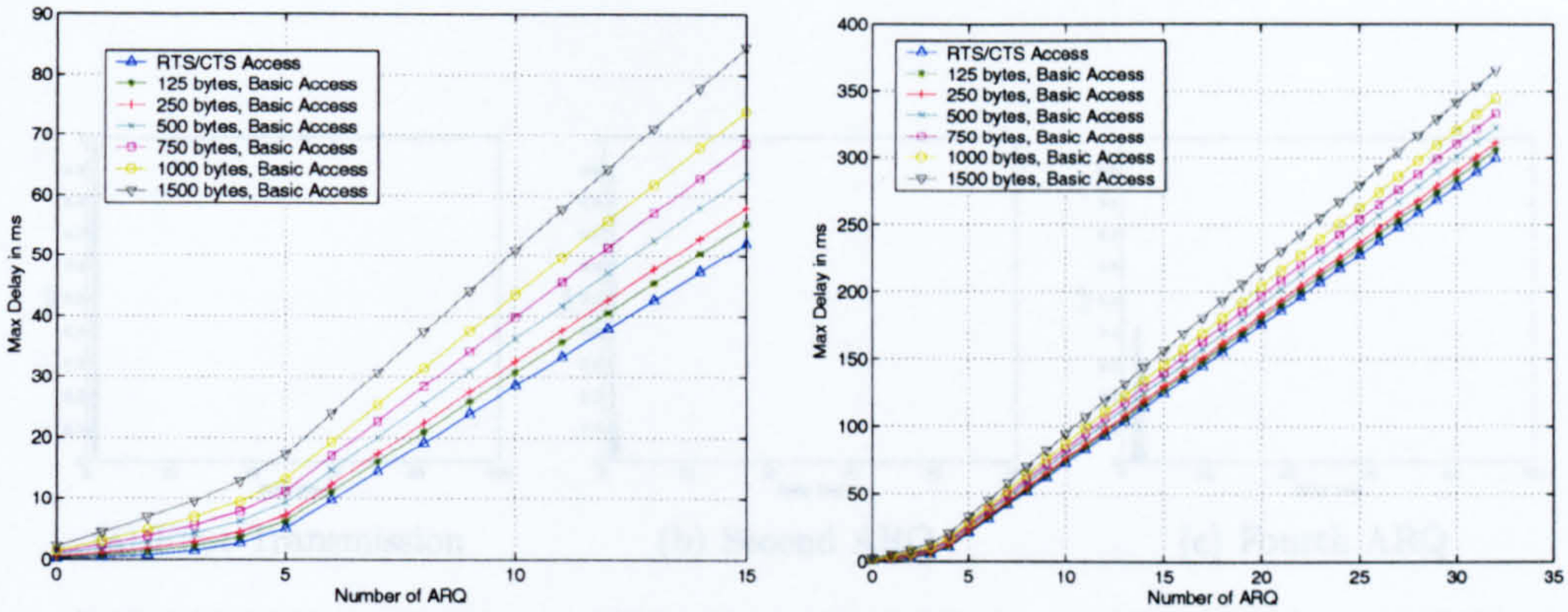
(c) 20Mbits/s

Mode	*Packet Length in bytes*
1	-
2	-
3	-
4	-
5	3000
6	1050
7	750

Moreover, it is assumed that retransmission occurs only on the RTS frame and not on the data frame. In the case of the basic access, the delay is packet length dependent since the station needs to transmit the data frame before knowing that the transmission has failed. Note that 0 ARQ corresponds to the first transmission. It can be seen that after 6 and 7 ARQs , the delay is linear for IEEE 802.11a and IEEE 802.11b/g respectively. This is because the maximum CW_{max} value is reached after 6 and 7 retransmissions respectively. Table 3.5 summarises the delays for the RTS/CTS and the basic access schemes for mode 1 for IEEE 802.11a. 16 ARQs introduces a delay of 51 ms before the pending packet is dropped. Such a delay is obviously not desirable for a single packet in video transmission. Table 3.6 highlights the difference in delays between IEEE 802.11a and IEEE 802.11b/g. Due to the larger DIFS and CW_{min}, delays are larger than the IEEE 802.11a case and 16 ARQ would double the packet delay for IEEE 802.11b/g parameters compared to 802.11a with mode 1. Note that RTS/CTS can lead to potential huge delays in the case where the RTS is received correct after several ARQs and the data frame is corrupted. The whole 'double' retransmission process restarts again with RTS, until the data frame is received correctly or the *maximum_retry_limit* is reached for either the RTS or data frame.

Table 3.5: DCF Delays for different numbers of ARQ (in ms) - IEEE 802.11a MAC parameters - Mode 1

Number of ARQ	*RTS/CTS Access*	*Basic Access 500 bytes*	*Basic Access 1000 bytes*
0 (1^{st} Transmission)	0.149	0.853	1.521
1	0.371	1.776	3.115
2	0.763	2.848	4.852
4	2.616	6.139	9.479
8	19.058	25.394	31.406
16	51.856	68.510	74.154



(a) IEEE 802.11a

(b) IEEE 802.11b/g

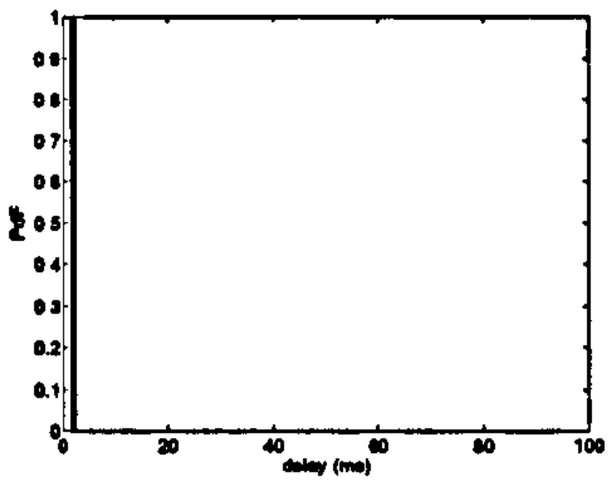
Figure 3.11: DCF Packet Delay Comparison - Mode 1

Table 3.6: DCF Delays comparison between 802.11a and 802.11b/g parameters for different number of ARQs (in ms) for a packet length of 1500 bytes - Mode 1

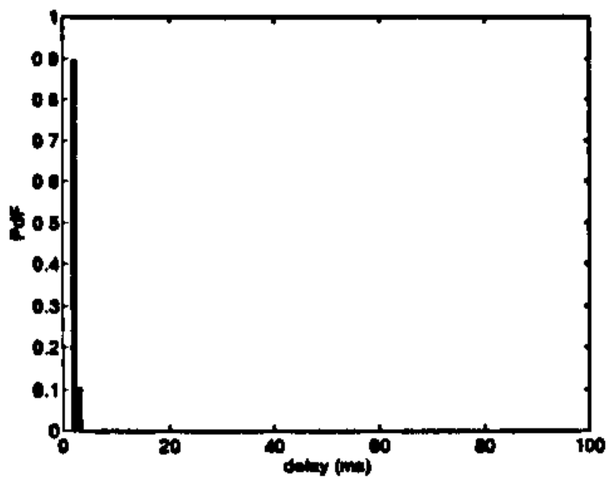
Number of ARQ	*Basic Access IEEE 802.11b/g*	*Basic Access IEEE 802.11a*
0 (1^{st} Transmission)	2.42	1.521
1	5.16	3.115
2	8.55	4.852
4	17.10	9.479
8	69.80	31.406
16	168.53	74.154

Figure 3.12 depicts the probability density function of the delay for packets encountering different numbers of retransmission with the RTS/CTS access. As the number of ARQ increases, the probability of packet delay spreads and shifts to the right to higher values due to the larger CW. This is confirmed by figure 3.13 which shows the cumulative probability density function of a packet having a delay smaller than the x-coordinate. The probability of having a delay smaller than 10 ms is 1 for the first transmission, and for the first, second, third and fourth ARQs. However, when seven and eight ARQs are used, this probability drops to 0.2 and 0.05 respectively.

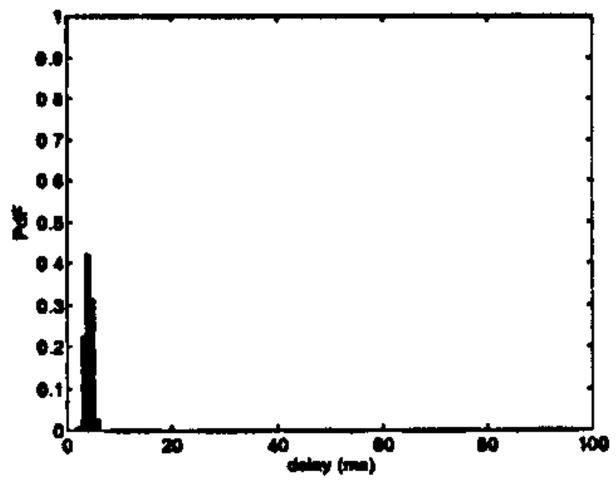
Figure 3.14 shows the influence of the initial (CW_{min}) and the maximum (CW_{max}) values of CW on the delay. From figure 3.14(a), where $CW_{max} = 1023$, it can be seen that the lower CW_{min}, the smaller the delay for a given number of ARQ. With a lower CW_{min}, the initial average back-off is then smaller. The maximum back-off stage m decreases as CW_{min} increases from m=6 for $CW_{min} = 15$ to m=*1* for $CW_{min} = 1023$. Figure 3.14(b),where $CW_{min} = 15$,shows that the larger CW_{max}, the larger the delay for a given number of ARQ. The maximum back-off stage m increases as CW_{max} increases from m=1 for $CW_{max} = 15$,to m=6 for $CW_{max} = 1023$.



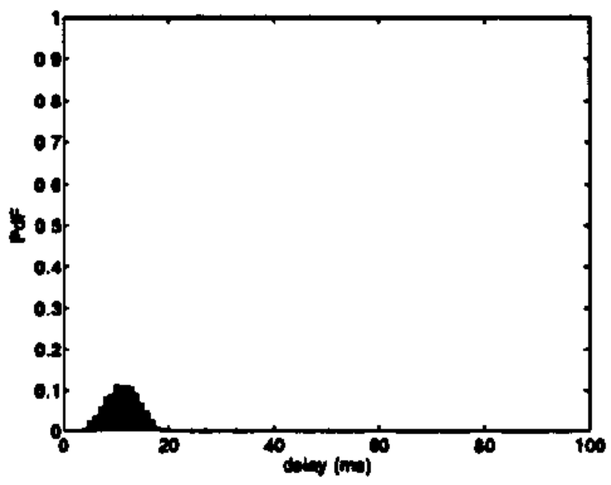
(a) First Transmission



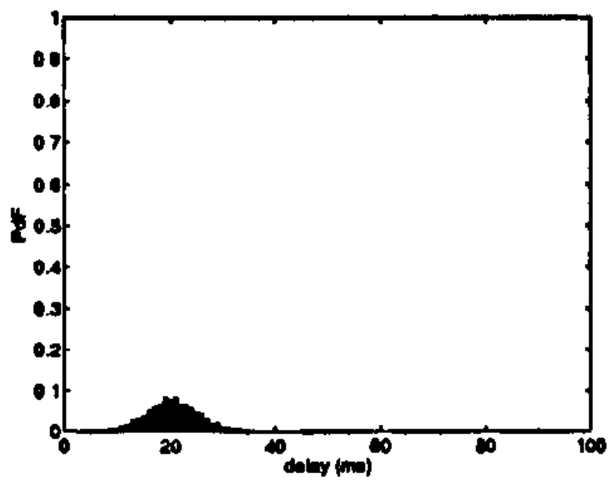
(b) Second ARQ



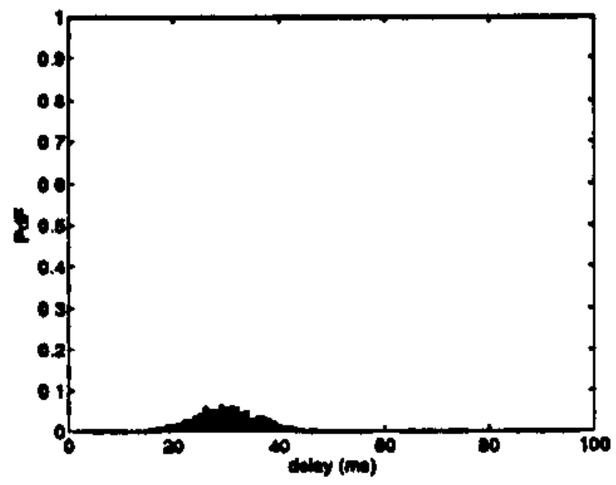
(c) Fourth ARQ



(d) Sixth ARQ



(e) Eighth ARQ



(f) tenth ARQ

Figure 3.12: Probability Density Function of Delay - RTS/CTS

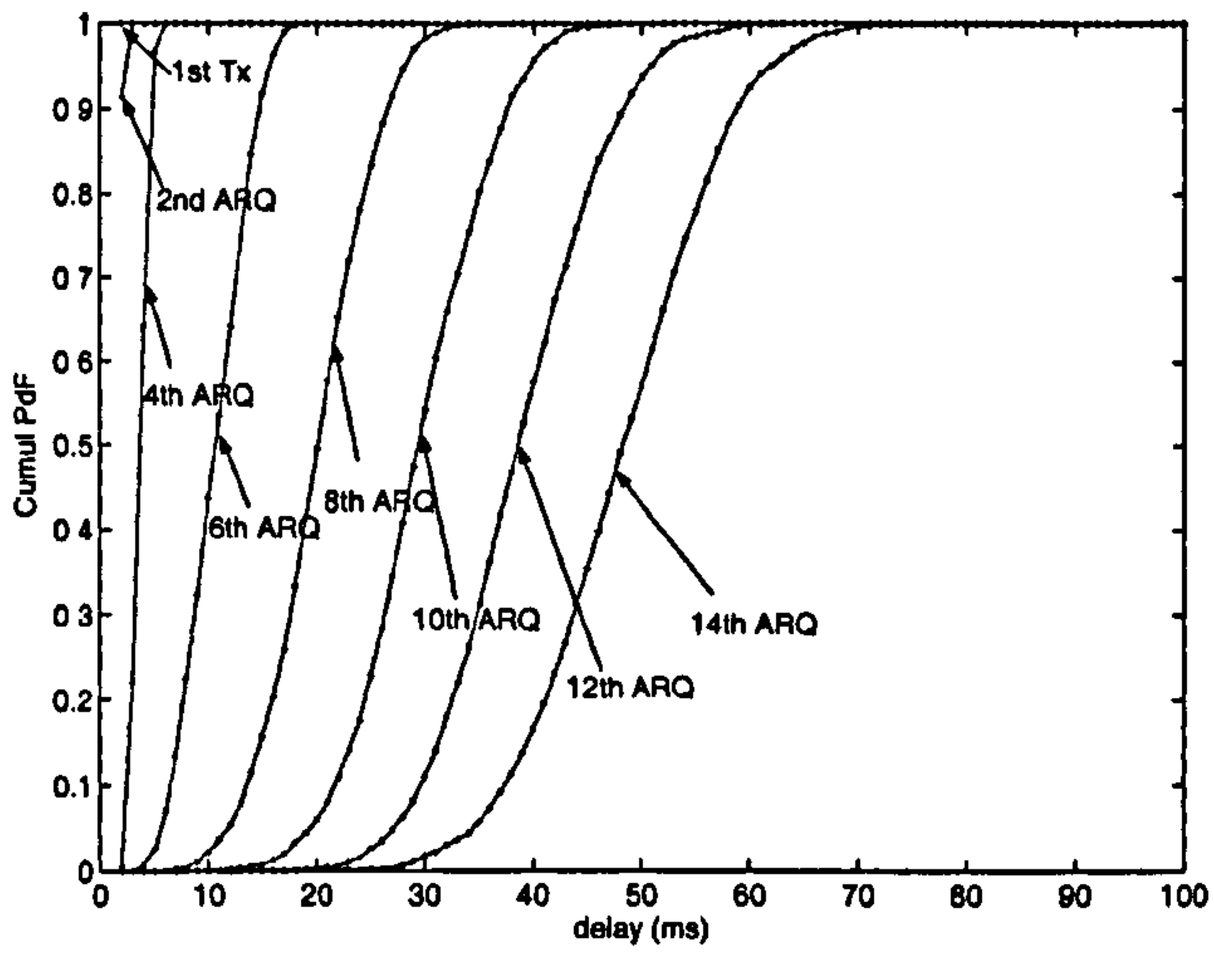
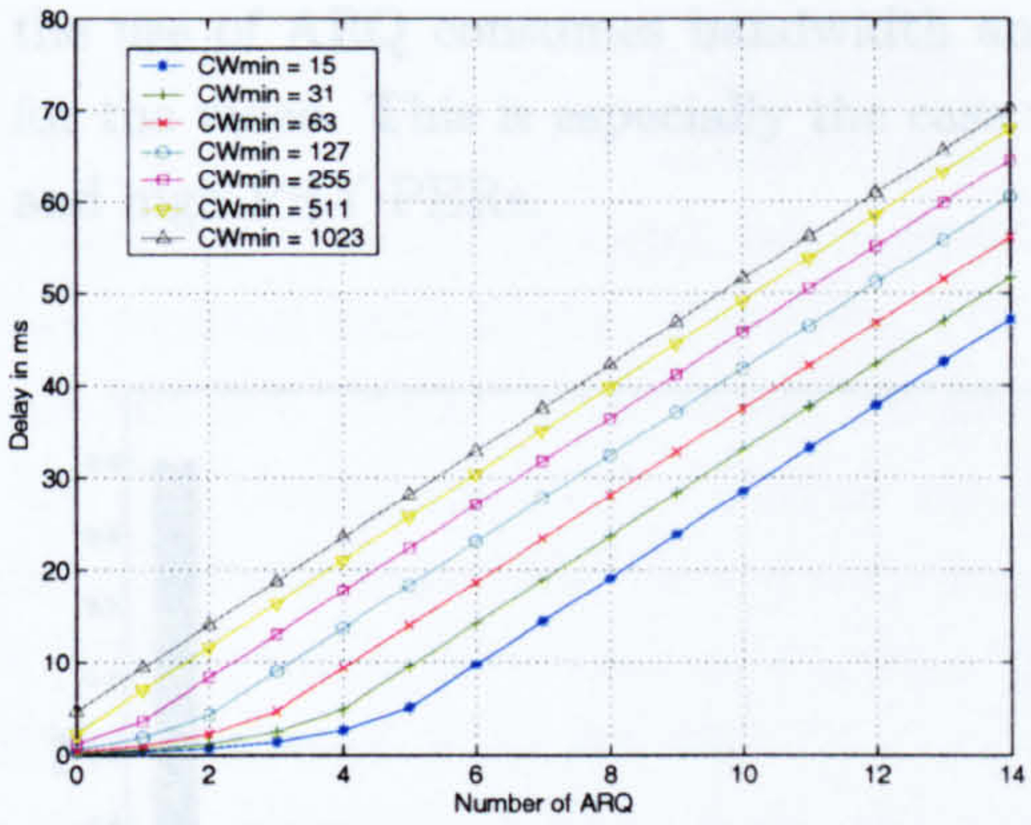
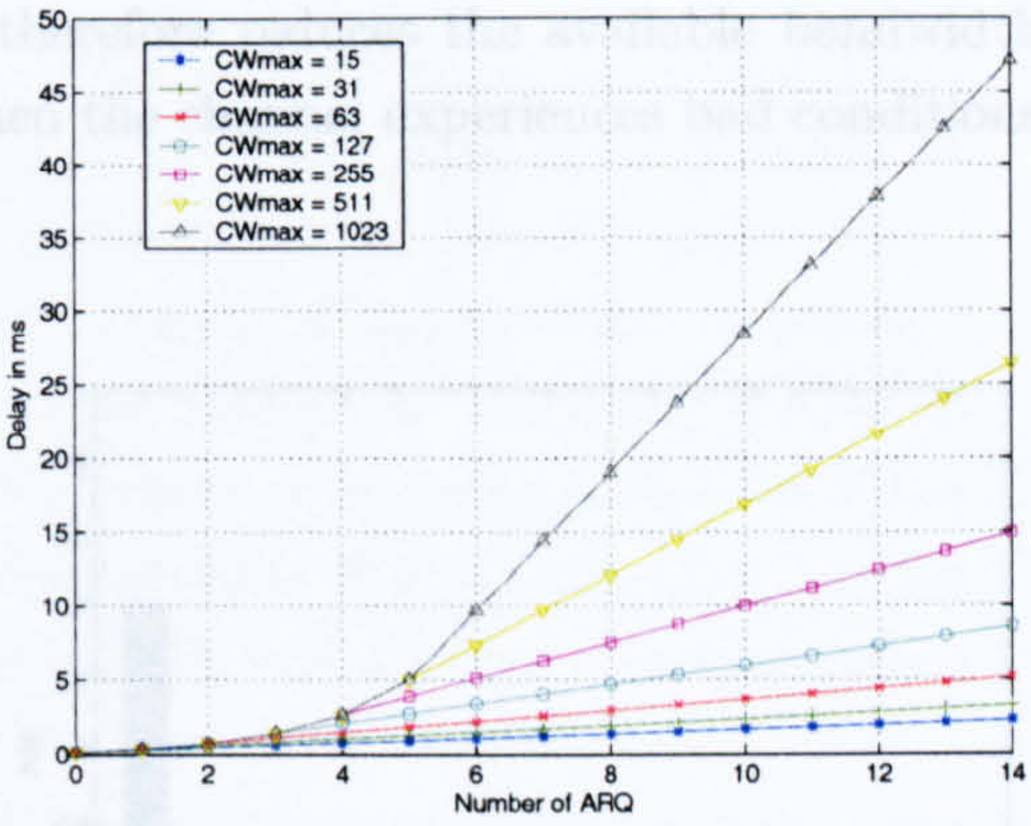


Figure 3.13: Cumulative Probability Density Function of Delay



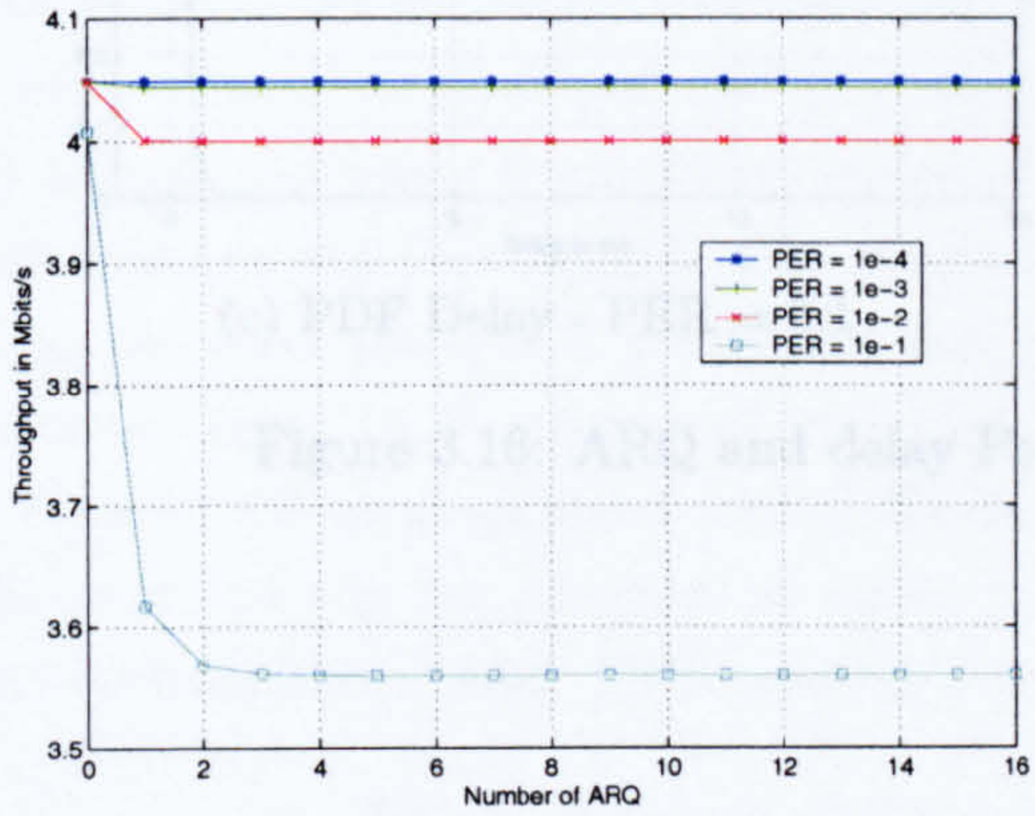
(a) Influence of Minimun CW



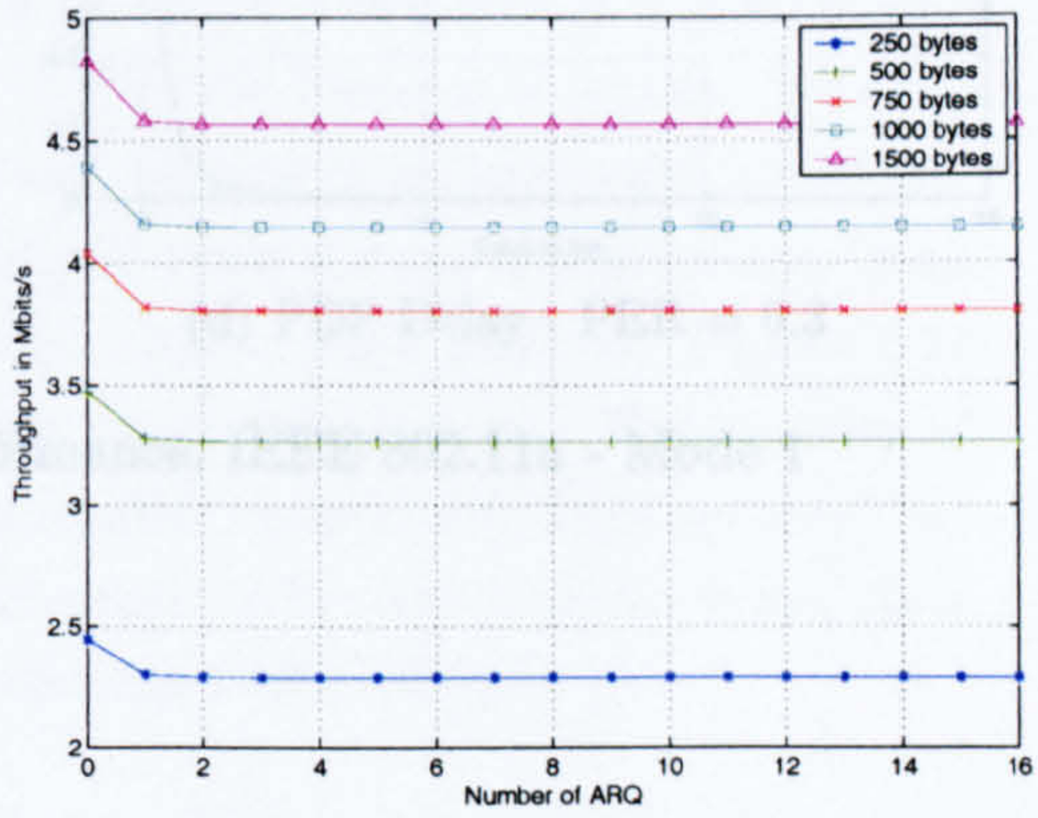
(b) Influence of Maximum CW

Figure 3.14: Influence of the Minimum and Maximum CW on the delay

Figure 3.15 shows the throughput variations for different number of ARQs and for different packet sizes for mode 1 with a PER of 5% (figure 3.15(a)) and for a packet length of 750 bytes (figure 3.15(b)) for the mode 1 of IEEE 802.11g. It can be seen that as the number of ARQ increases, the delivery time increases and the throughput decreases. However, with low PER, the reduction in throughput is small since only a few packets require retransmission and the bandwidth used for retransmission is small.



(a) Packet Length = 750 bytes

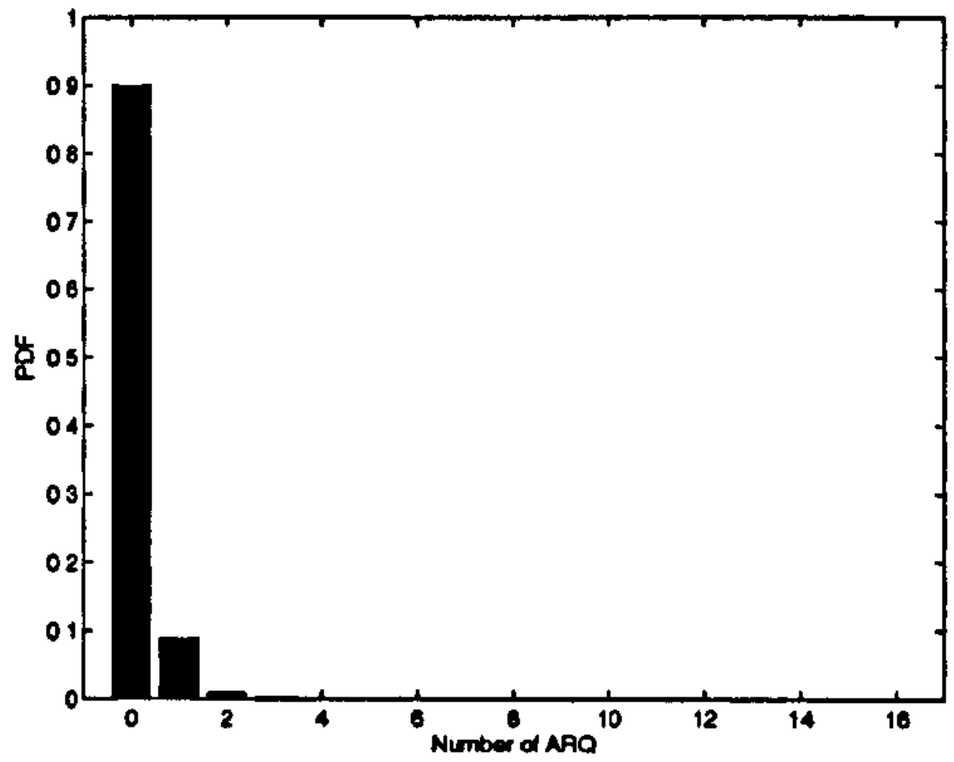


(b) PER = 5%

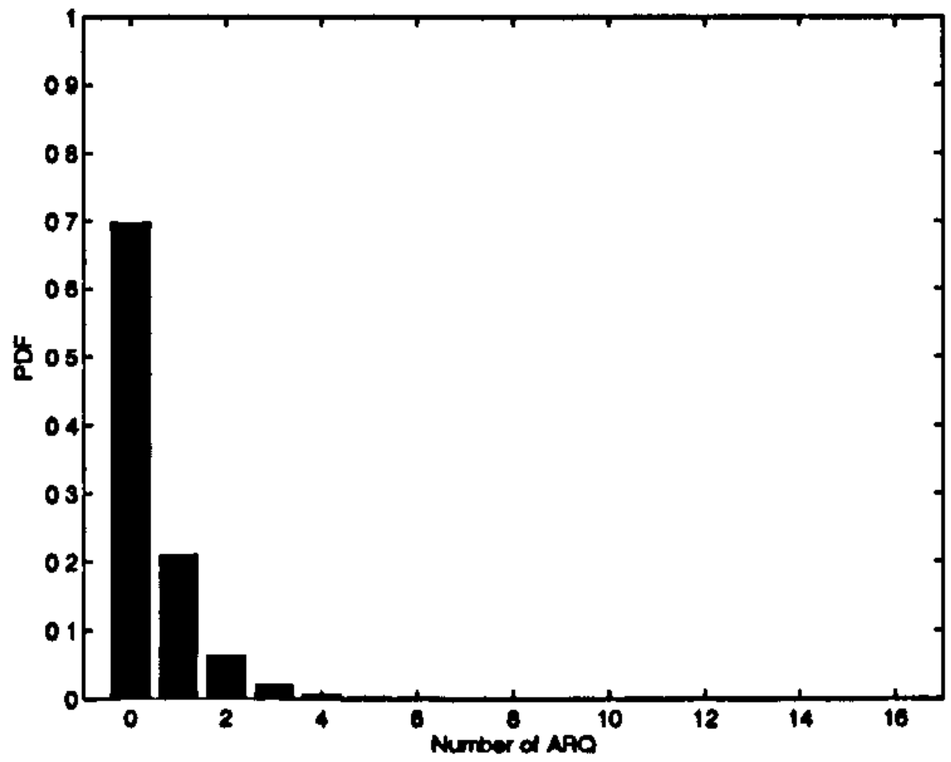
Figure 3.15: DCF Throughput Comparison versus Number of ARQ- IEEE 802.11g - Mode 1

There is a fundamental trade-off between PER, throughput and delay for video transmission and the number of ARQs implemented at the MAC layer is crucial for a real-time video application. Figure 3.16 shows the PDF of number of ARQs and PDF of packet delay for a PER of 0.1 and 0.3 with mode 3 of the IEEE 802.11 a PHY. As the PER increases, the ARQ mechanism is called more often. Using ARQ obviously reduces the PER at the MAC layer, and therefore provides better visual quality. However, using ARQ introduces end to end delays. Remaining packets are then delayed and large delays

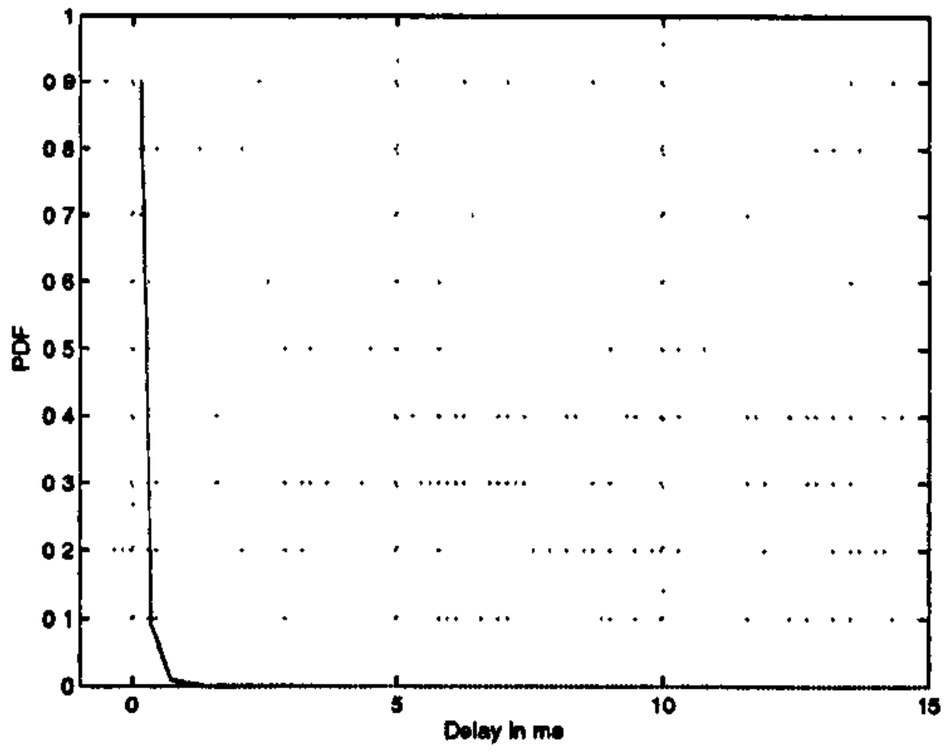
are building up that are not suitable for time-bounded video applications. Moreover, the use of ARQ consumes bandwidth and therefore reduces the available bandwidth for the video. This is especially the case when the channel experiences bad conditions and high PHY PERs.



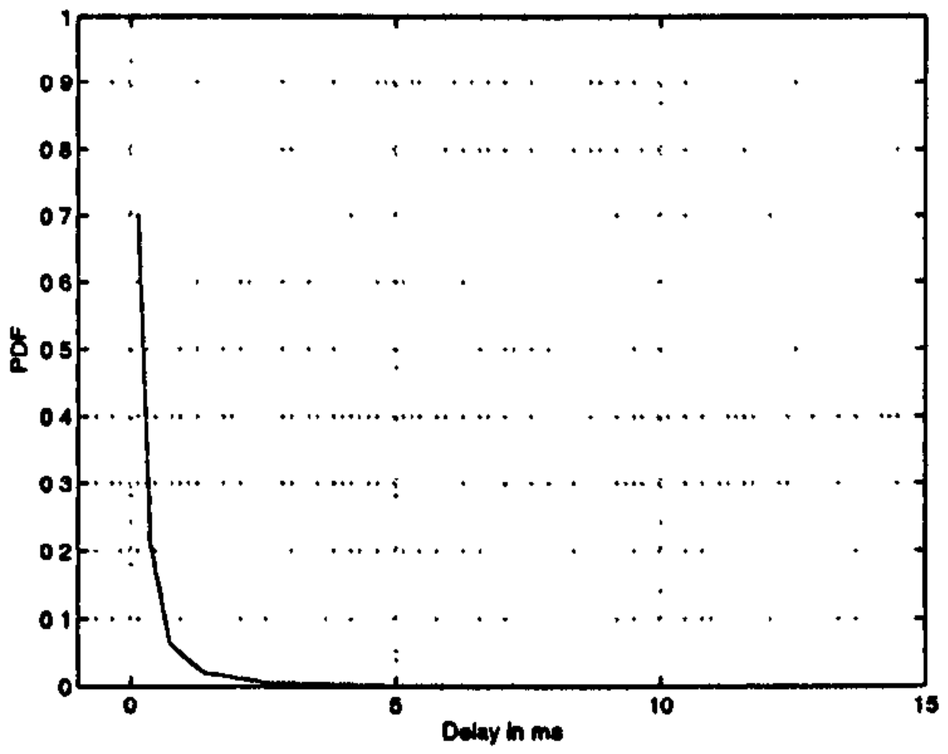
(a) PDF ARQ - PER = 0.1



(b) PDF ARQ - PER = 0.3



(c) PDF Delay - PER = 0.1



(d) PDF Delay - PER = 0.3

Figure 3.16: ARQ and delay Performance, IEEE 802.11a - Mode 1

3.3 Point Coordination Function

The optional PCF has been designed to support time-bounded services and can only be deployed on infrastructure network configurations. It provides a contention free (CF) period (CFP) for transmission by implementing a polling access method. However, it relies on the asynchronous access of the DCF [39]. Because PCF is thought to be too complex [75], PCF has only been implemented in a few chipsets and a short study is presented here. The simulations performed and the results presented in this section assume a 'lightly loaded network'.

3.3.1 Access Mechanism

This access method uses a Point Coordinator (PC) also known as an Access Point (AP). The PC polls and coordinates stations and lets them have priority access to the medium. It therefore eliminates contention among stations. The PC gains control of the medium periodically. Once the PC gains control of the medium, it begins a contention free period during which the access to the medium is completely controlled by the PC. During the PCF, a station can only transmit after being polled. PCF has a higher priority than DCF since it may start transmissions after a Point Coordination IFS (PIFS), which has a shorter duration than a DIFS and a longer duration than a SIFS (see equation 3.11) [27, 80].

PCF starts with a beacon frame [39, 81]. A beacon frame is a management frame that maintains the synchronisation between stations and delivers timing related parameters. Beacon frames are periodically delivered at each Target Beacon Transition Time (TBTT). The TBTT defines the time duration before the next beacon frame [85]. A beacon frame notifies all the other stations not to initiate transmission for the length of the CFP. The PC can then allow a given station to have contention-free access through the use of a polling frame (CF-poll frame) [86]. The CF-poll frame may contain data if the PC has pending data to be transmitted to the station. The beacon frame can also contain association/disassociation frame information as well as authentication information. The polling ends with a CF-End frame sent by the PC, or at the time announced in the beacon frame

After the initial beacon frame, the PC shall wait for at least one SIFS before transmitting one of the following:*i*) a data frame, *ii*) a Contention-Free poll (CF-poll) Frame or *iii*) a Contention-Free poll and data frame. A CF-poll frame is a request from the PC to poll stations. Upon being polled, stations acknowledge successful reception. Depending on the PCF length, and the data length, several data transmissions (with ACK) can take place in a contention free period after a SIFS interval. Figure 3.17 shows the basic PCF access with only one data frame transmitted. The priority of PCF over DCF is guaranteed with PIFS being smaller than DIFS. This ensures that other stations will not attempt to access the channel. After the beacon frame, data shall be transmitted after SIFS, and then acknowledged after SIFS.

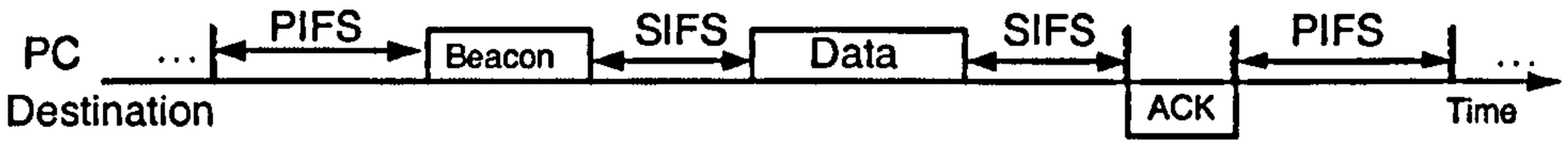


Figure 3.17: PCF Access Mechanism

3.3.2 Throughput Performance

Based upon figure 3.17, a successful transmission cycle is given by equation 3.14

$$T_{success}^{PCF} = PIFS + Beacon + SIFST_{Data} + SIFS + ACK \tag{3.14}$$

The beacon frame duration depends on the content of the body frame (timing, association, authentication parameters) and it is assumed here to be $36\mu s$ [81]. The throughput is derived in the same way as with the DCF. Figure 3.18 shows the throughput performance for different packet lengths and for the operating mode of the IEEE 802.11a PHY. As with DCF, due to the smaller relative MAC overhead, the throughput increases as the packet length increases. By comparing figures 3.7(a) and 3.18, we can see that PCF offers slightly better performance over Basic DCF since there is no need for a back-off (no contention) and since a station has to wait for a smaller time interval (PIFS is smaller than DIFS). However, performance of both schemes highly depends on the packet length.

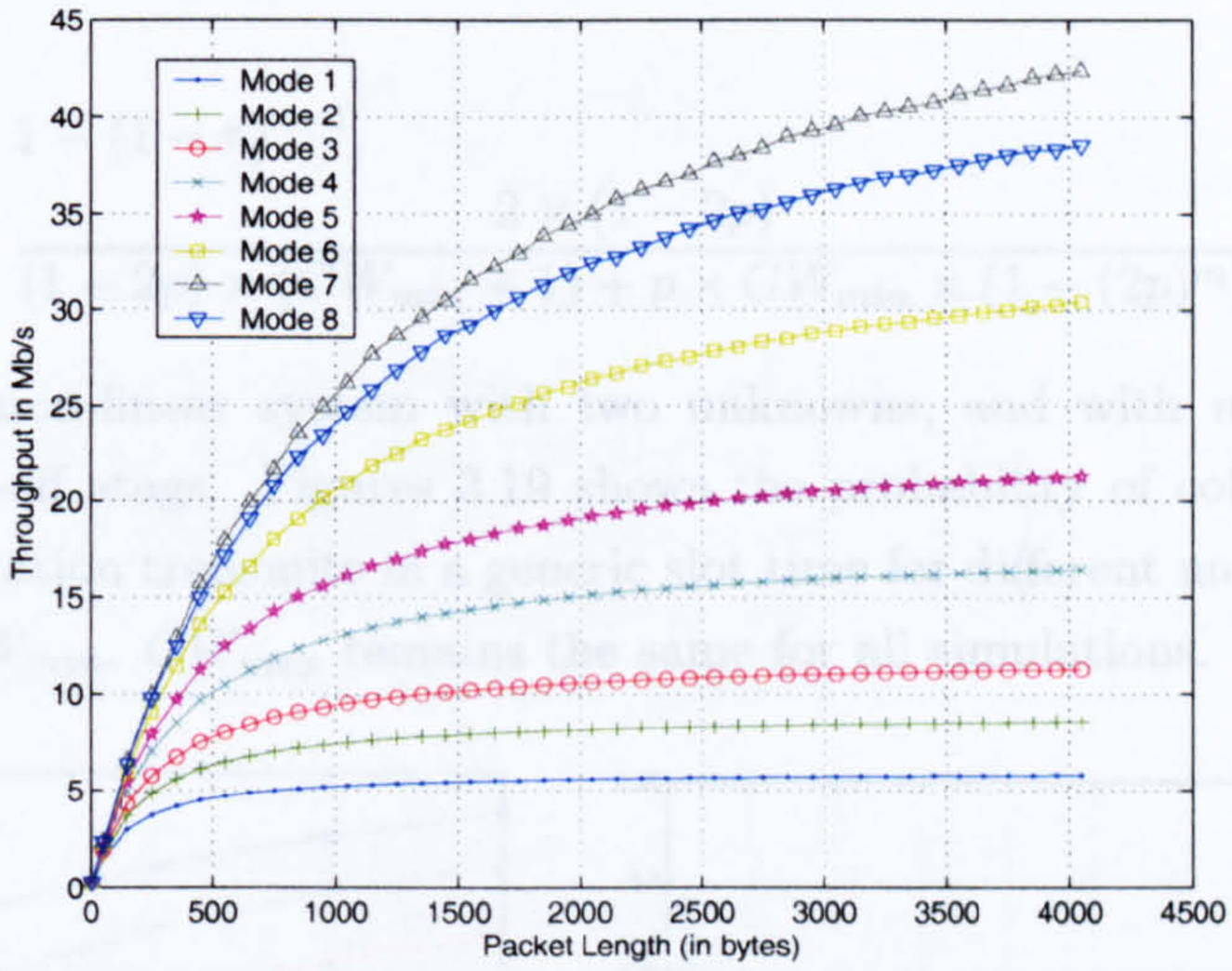


Figure 3.18: Throughput PCF - Basic Access

3.4 Throughput and Delay Analysis with Markov Chains

In section 3.2, it was assumed that stations have access to the medium after the first attempt and there were no collisions. In the case of several users, typically more than 3, this assumption is no longer valid. The random back-off can not be taken as the average and the probability of collision is not zero anymore. In case of collision, the contention window follows an exponential increase up to the maximum. The derivation of all the equations presented in this section is detailed in Appendix C. This study reproduces the work presented in [25, 87], but is applied to a IEEE 802.11a PHY layer.

3.4.1 Probabilities of Transmission and Collision

In [25, 87], Bianchi developed a theory based on Markov Chains and stochastic processes to analyse the performance of the IEEE 802.11 DCF MAC taking into account the probability of collision with different numbers of users. He assumed that the probability of collision is constant and independent for each transmitted packet, that the channel is ideal without any hidden terminals, and that the number of stations is fixed during the

simulation. He also assumed that the transmission queue of each station is non-empty (saturation condition). This theory has also been studied in [88] and [89] with an IEEE 802.11b PHY. Here, it is applied it to the IEEE 802.11a PHY and the DCF access scheme is considered.

Let us define:

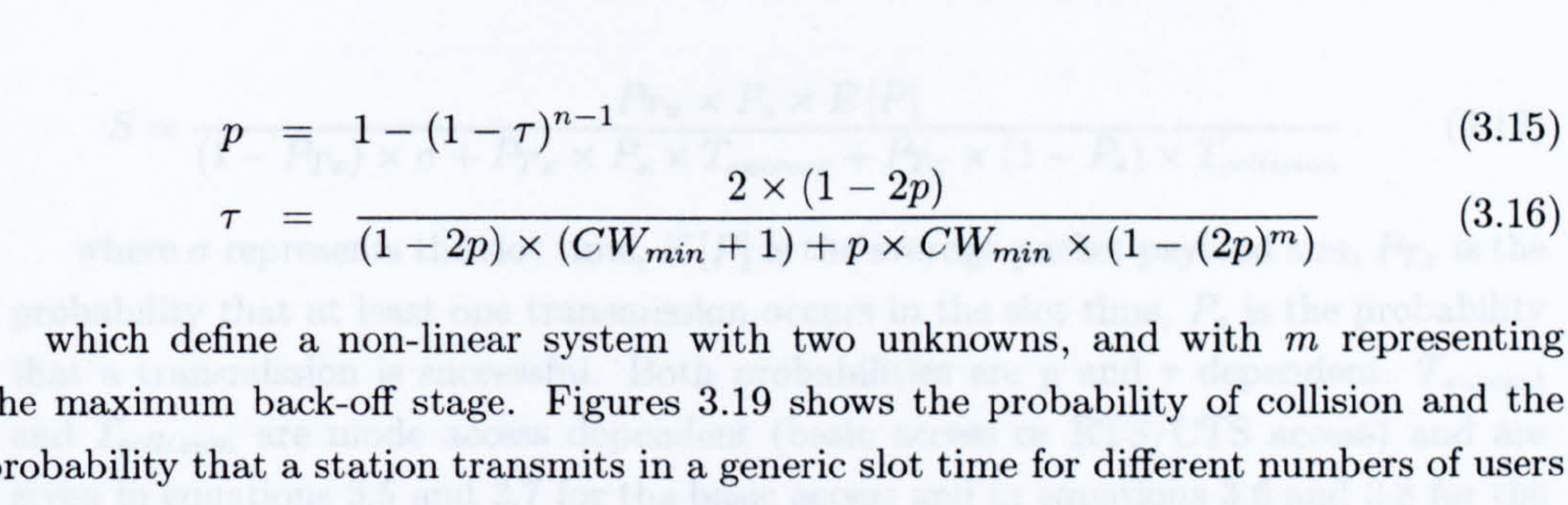
- p: the probability of collision seen by a packet being transmitted on the channel.
- τ: the probability that a station transmits a packet in a generic slot time.

τ and p can respectively be derived as:

$$p = 1 - (1 - \tau)^{n-1} \tag{3.15}$$

$$\tau = \frac{2 \times (1 - 2p)}{(1 - 2p) \times (CW_{min} + 1) + p \times CW_{min} \times (1 - (2p)^m)} \tag{3.16}$$

which define a non-linear system with two unknowns, and with m representing the maximum back-off stage. Figures 3.19 shows the probability of collision and the probability that a station transmits in a generic slot time for different numbers of users and for different CW_{min}. CW_{max} remains the same for all simulations.



(a) Probability of Collision

(b) Probability of Transmission in a generic slot time

Figure 3.19: Probability of Collision and Transmission

From figure 3.19(a) it can be seen that as the number of users increases, the probability of collision increases. As CW_{min} increases, the probability of collision decreases. This is explained by the fact that there are more slot times available as CW_{min} gets larger, so it is less likely for two stations to have their counters equal to zero at the same time. With 10 users, a CW_{min} of 7 leads to a probability of collision of almost 0.5. Using a CW_{min} of 63 reduces the probability of collision to 0.2.

Figure 3.19(b) shows that the probability that a station transmits in a generic slot time decreases as the number of users increases. This is explained by the fact that there are more stations contending the channel, so each of them has less chance to transmit.

When CW_{min} increases, as there are more slot times before the counter reaches zero, the probability of transmission decreases as well [25, 80]. Note that the IEEE 802.11a PHY parameters for the MAC are m=6 and $CW_{min} = 15$.

3.4.2 Throughput and Delay Performance

The normalised system throughput, defined as the fraction of time used to successfully transmit payload bits, is given by:

$$S = \frac{E[Payload\ Tx\ in\ a\ Slot\ Time]}{E[Length\ of\ a\ Slot\ Time]} \tag{3.17}$$

and can be expressed as (see Appendix C):

$$S = \frac{P_{Tx} \times P_s \times E[P]}{(1 - P_{Tx}) \times \sigma + P_{Tx} \times P_s \times T_{success} + P_{Tx} \times (1 - P_s) \times T_{collision}} \tag{3.18}$$

where σ represents the slot time, $E[P]$ is the average packet payload size, P_{Tx} is the probability that at least one transmission occurs in the slot time, P_s is the probability that a transmission is successful. Both probabilities are p and τ dependent. $T_{success}$ and $T_{collision}$ are mode access dependent (basic access or RTS/CTS access) and are given in equations 3.5 and 3.7 for the basic access and in equations 3.6 and 3.8 for the RTS/CTS access respectively. P_{Tx} and P_s are given by:

$$P_{Tx} = 1 - (1 - \tau)^n \tag{3.19}$$

$$P_s = \frac{n \times \tau \times (1 - \tau)^{n-1}}{1 - (1 - \tau)^n} \tag{3.20}$$

Figure 3.20 shows the normalised system throughput for the two DCF access schemes for different numbers of users, and different packet lengths, under mode 1 of the IEEE 802.11a PHY-layer and ETSI-Channel A. This is also summarised in table 3.7. Note that in the case of RTS/CTS, it is assumed that the collision and therefore retransmission only occurs on the RTS frame and not in the data frame.

Table 3.7: Normalised Throughput Comparison

Number Users	128 bytes basic	128 bytes RTS/CTS	512 bytes basic	512 bytes RTS/CTS	1024 bytes basic	1024 bytes RTS/CTS
1	0.570	0.430	0.805	0.715	0.885	0.831
5	0.541	0.446	0.721	0.728	0.775	0.838
10	0.509	0.446	0.667	0.728	0.712	0.838
20	0.474	0.443	0.612	0.723	0.652	0.836
40	0.436	0.437	0.556	0.721	0.589	0.833

The basic access throughput decreases rapidly as the number of users increases

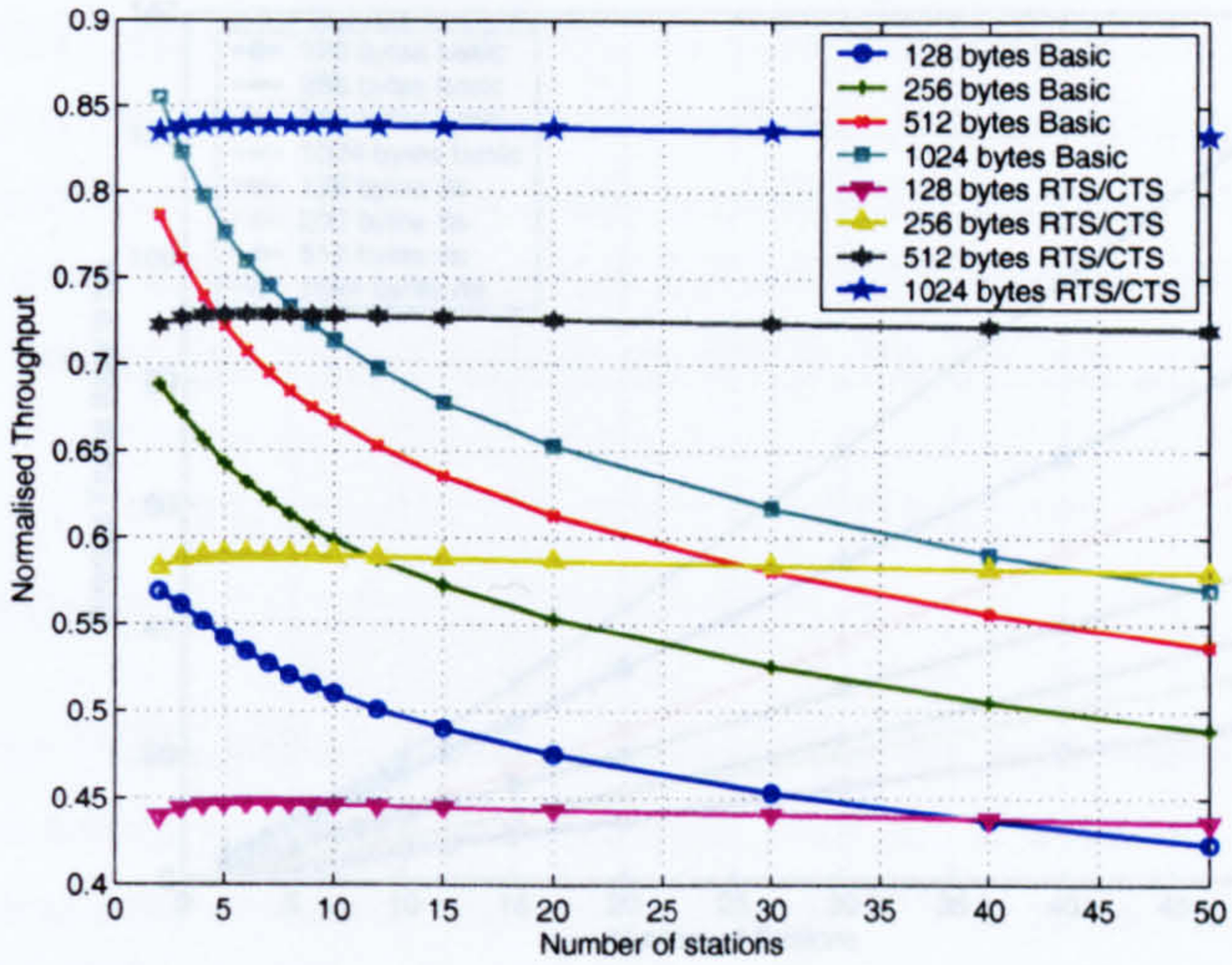


Figure 3.20: Normalised Throughput

compared to the RTS/CTS throughput, which remains almost constant. This is due to the fact that the probability of collision increases. With the RTS/CTS scheme, only the first RTS frame is lost, whereas in the basic scheme the whole data frame is lost. The difference is seen to be more important with long packets. For a small number of users, the RTS/CTS scheme uses some of its bandwidth for RTS frames, where the collision probability is low. It therefore does not make good advantage of the channel resources. With longer packets, when a collision occurs, only the short RTS frame is lost with the RTS/CTS scheme, whereas the whole data frame is lost with the basic scheme. With 1024 byte long packets and with 10 users, the offered throughput is 11% higher for the RTS/CTS scheme compared to the basic scheme. On the other hand, with 128 byte long packets, the offered throughput is 5% higher for the basic scheme. It is therefore recommended to use RTS/CTS for long packets and to use the basic scheme for small packets [80]. The IEEE 802.11 MAC defines a threshold above which packets are transmitted with RTS/CTS. The value of this threshold is chosen between 0 and 2347 bytes, but is not standardised and is left to manufacturers.

The average frame delay is given by:

$$E[D] = E[x] \times E[Length\ of\ a\ Slot\ Time] \tag{3.21}$$

where $E[x]$ is the average number of slot times required for successfully transmitting a new frame. $E[D]$ is developed in appendix C.

Figure 3.21 shows the average frame delay under the same PHY and channel conditions. It can be seen that the RTS/CTS scheme provides better delay performance as the number of users increases, since only the RTS frame is lost in the case of collision, whereas the basic scheme loses the whole data frame.

The use RTS/CTS reduces the probability of collision. It provides better throughput performance when compared to the basic access, and therefore, in a multimedia system where severals users are accessing video content, the reduction of available band-

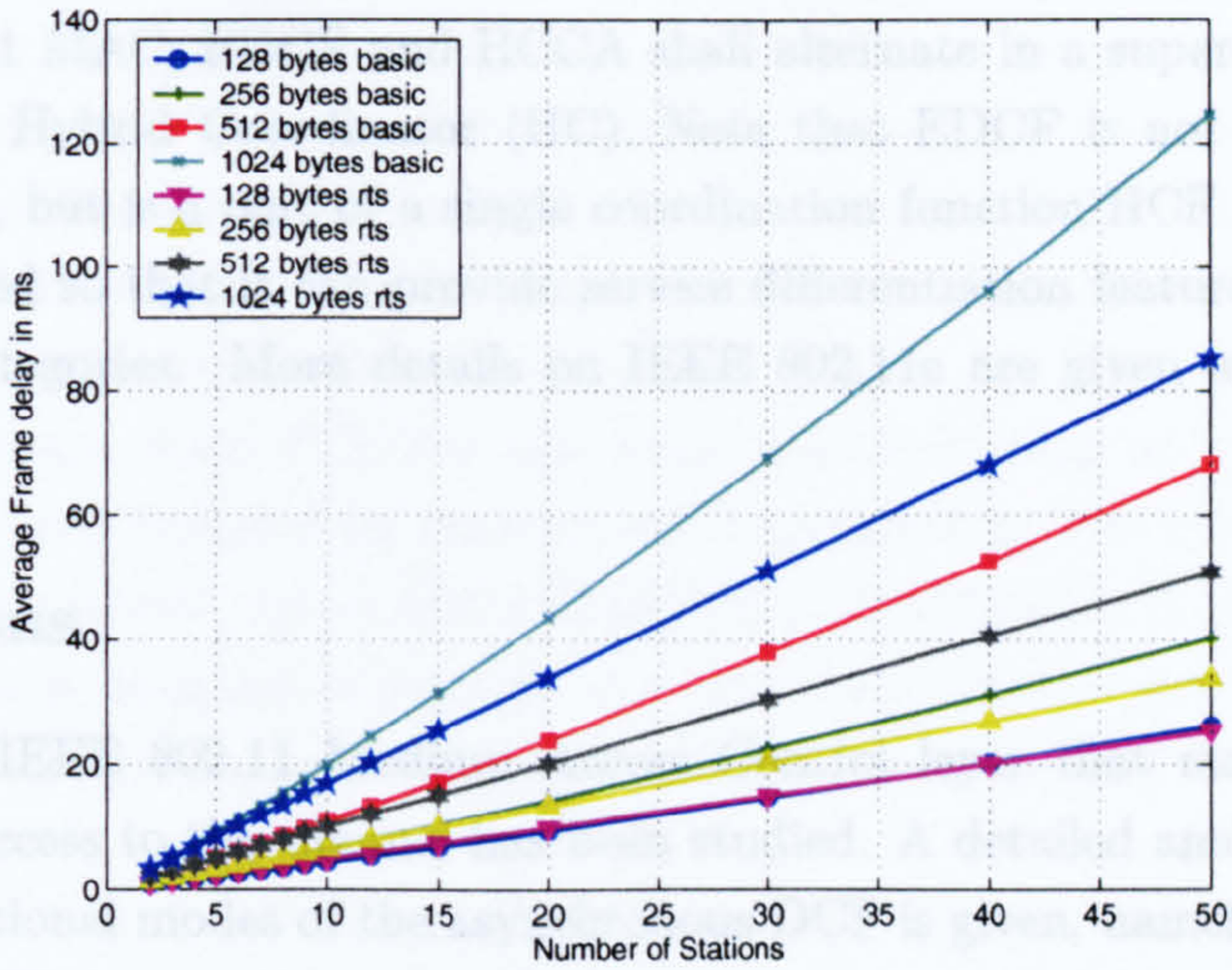


Figure 3.21: Average Frame Delay

width for video application is limited. However, as number of users increases, even if RTS/CTS slightly reduces the delay compared to the basic scheme, large delays are noticeable which is not compliant with time-bounded application requirements. The number of users accessing a real-time multimedia distribution system should therefore be limited.

3.5 IEEE 802.11 Quality of Services Limitations

Even though the IEEE 802.11 PCF has been designed to support time-bounded services, there are problems that constrain its use. One of them is the unpredictable beacon delay due to the unknown transmission time of the polled stations. At TBTT (see section 3.3), the PC schedules the beacon as the next frame to be transmitted, but it can only be transmitted if the medium has been sensed as idle for at least PIFS. The medium may not be idle at this time and the beacon frame would then be delayed. This would automatically delay the time-bounded data frames that were set to be sent under the CF mode. Another problem is the unknown duration of the data frame transmitted by the polled stations. These frames may have variable lengths and can be sent with different transmission modes. This can not be controlled by the PC [23, 85]. Moreover, data transmitted during CP with DCF is not controlled by the PC and this can delay further time bounded transmissions. In addition, the time each user has to wait just to access the channel might be unacceptable for real-time transmission if the number of users increases.

In order to support QoS, the IEEE 802.11 Task Group E is currently developing an enhanced version of the IEEE 802.11 MAC called IEEE 802.11e [40]. This enhanced version defines the Hybrid Coordination Function (HCF) to access the channel. HCF uses the Enhanced DCF (EDCF) also called Enhanced Distributed Channel Access (EDCA) [23] for the contention access period, CP, and the Hybrid Coordination Chan-

nel Access (HCCA) for the controlled access and contention free period, CFP. As in the legacy IEEE802.11 MAC, EDCF and HCCA shall alternate in a super-frame and are coordinated by a Hybrid Coordinator (HC). Note that EDCF is not a separate coordination function, but is a part of a single coordination function HCF [38]. IEEE 802.11e is also designed so that it can provide service differentiation features with parameterised access categories. More details on IEEE 802.11e are given in appendix D.

3.6 Conclusions

In this chapter, the IEEE 802.11 Medium Access Control layer that manages and provides the shared access to the channel has been studied. A detailed analysis of the two mandatory operational modes of the asynchronous DCF is given, namely the basic mode and the RTS/CTS access schemes, including throughput and delay studies. The throughput at the MAC layer is packet length dependent. The IEEE 802.11 MAC has a very poor overhead efficiency. Due to smaller overheads and better use of channel resources, long packets provide better throughput than small packets. Packets of 500 bytes have a 25% overhead, where 1500 byte long packets have only a 10% overhead with mode 1 of IEEE 802.11a. In order to be able to transmit a video encoded at a specific bit rate, video packets are required to have a minimum length. Long packets at the MAC might therefore be preferable for high bit rate applications such as video applications. Moreover, due to channels error, the MAC throughput is further reduced.

The stop and wait ARQ is a mandatory feature of the IEEE 802.11 MAC. It is used to retransmit corrupted packets but also to detect collision. It uses a back-off procedure to avoid collision. This however introduces long packet delays. A packet would suffer 20 ms delay if it requires to be retransmitted 8 times with the RTS/CTS scheme using IEEE 802.11a parameters. Such delays might not be acceptable for real-time services. Packets in the transmission queue would also suffer the delay. This delay builds up when several packets are delayed. The maximum number of ARQs allowed is not specified by the standard and is left to the manufacturers and/or the applications. The retransmission generated by the ARQ mechanisms consumes bandwidth and reduces the available throughput. A fundamental trade-off therefore appears between PER, throughput and delay that is crucial for time bounded applications. On one hand, the ARQ mechanism reduces the PER. On the other hand, it consumes bandwidth and delay the transmission. The use of ARQ should therefore be minimised, especially for time bounded video applications.

It has been demonstrated that IEEE 802.11g is backward compatible with b by using the IEEE 802.11b MAC parameters. These are larger than the IEEE 802.11a MAC parameters and generate larger overheads. IEEE 802.11g therefore offers smaller throughput performances. Similarly, the IEEE 802.11g delays are larger than those experienced by 802.11a.

An analysis of DCF with Markov Chains has been given in order to establish the influence of the number of users on the throughput and the delay. As the number

of users increases, the throughput decreases. However, the RTS/CTS provides better throughput performance than the basic access since collisions occur on the RTS frame and not on the data frame. However, even if the delay is reduced when using RTS/CTS compared to the basic access, as the number of users increases, delay is also increasing. The number of users accessing a multimedia distribution system should therefore be limited in order to guarantee reasonable end-to-end delay.

The optional access mode PCF has also been presented along with a throughput analysis. PCF has been designed for time bounded applications. However, it still lacks QoS support. The enhanced version of the IEEE 802.11 MAC, called IEEE 802.11e, has been introduced. It is designed to support QoS and it provides service differentiation features with parameterised access categories. For the remaining of this thesis, only the DCF is considered.

Chapter 4

Video Coding Techniques

In this chapter, an introduction to video coding techniques and standards is presented. The chapter is organised as follows. Section 4.1 gives an overview of video coding techniques. The Moving Picture Experts Group (MPEG) 2 video coding standard is detailed in section 4.2. This section contains a description of the coding features of MPEG-2 and it also describes the system layer of MPEG-2. Section 4.3 gives an overview of existing techniques used to overcome transmission errors and their effect on the decoded video. The new H.264 standard jointly designed by ITU-T Video Coding Experts Group (VCEG) and ISO/IEC MPEG is presented in section 4.4. This section describes the new features for improving the coding efficiency and the error resilience together with a description of the network abstraction layer of the encoder. Finally, section 4.5 concludes this chapter.

4.1 Introduction: Basics of Video coding

Video signals comprises high volumes of data are that not well suited to real time transmission.Video compression is thus required to reduce this high amount of data. It involves the extraction of pertinent information and the elimination of redundancy [90].

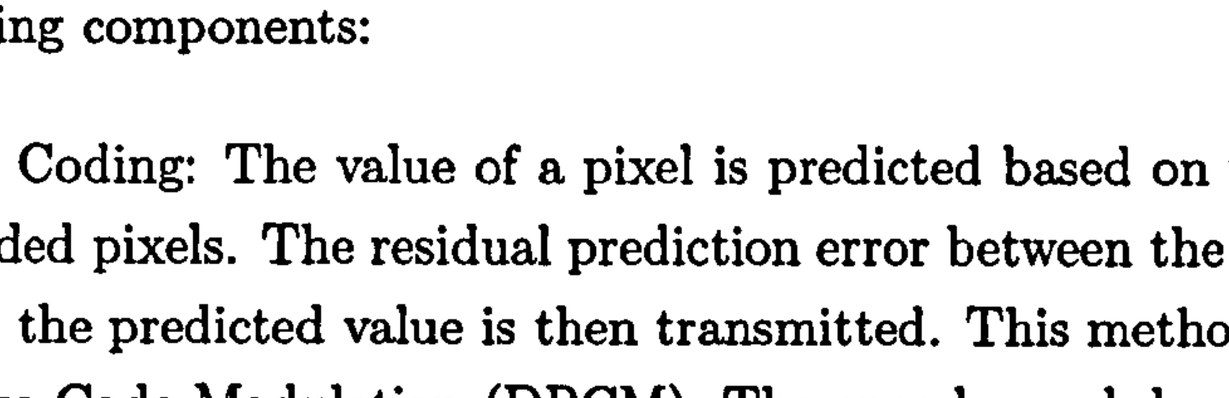
In a video sequence, there is a strong correlation between both successive frames (temporal redundancy) and within the picture elements (spatial redundancy). In theory, de-correlation leads to bandwidth compression without affecting image resolution. Moreover, the insensitivity of the Human Visual System (HVS) to the loss of certain spatio-temporal visual information can be exploited for coding [91]. For example, there is no need to transmit information that the human eye cannot see. In addition, it is possible to use lossy compression techniques, i.e. where some information are irreversibly lost, to reduce video bit-rate while maintaining an acceptable image quality. Considering still images, only spatial correlation is exploited. This type of compression is called Intra frame coding. On the other hand, if temporal correlation is exploited, then the compression technique is characterised as Interframe coding. The main video encoding scheme used in all standard video codecs is called Predictive Inter frame Coding. It uses three main reduction principles [90]:

- Spatial redundancy reduction among the pixels within the pictures (intra frame coding)
- Temporal redundancy reduction to remove similarities between pictures (inter frame coding)
- Entropy coding to reduce the redundancy between compressed data symbols

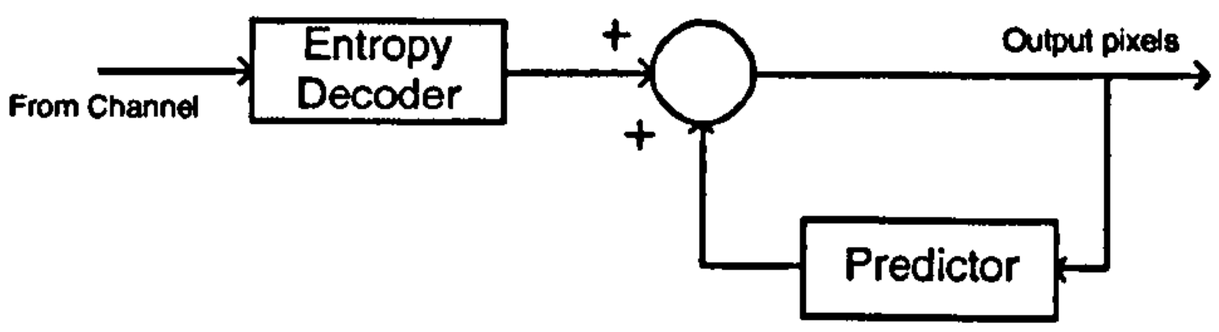
4.1.1 Spatial Redundancy Reduction

Spatial redundancy reduction refers to intra-frame coding (as in still image) and uses the three following components:

- Predictive Coding: The value of a pixel is predicted based on the values of previously coded pixels. The residual prediction error between the values previously coded and the predicted value is then transmitted. This method is called Differential Pulse Code Modulation (DPCM). The encoder and decoder are shown in figure 4.1. The difference between the incoming pixel and the predicted pixel is quantised and entropy coded prior to the transmission. Best predictions are taken from the neighboring pixels either from the same frame (intra frame predictive coding) or from the previous frame (inter frame predictive coding) or from both combinations (hybrid predictive coding).



(a) Encoder



(b) Decoder

Figure 4.1: Block Diagram of a DPCM Codec

- Transform Coding: The aim of this method is to de-correlate data and to concentrate the energy in a reduced number of coefficients. It removes spatial redundancies in images by mapping pixels onto a transform domain prior to data reduction. This method concentrates the image energy in the low frequency region, and therefore into a few transformed coefficients. A subsequent quantisation process can discard insignificant coefficients and provide good energy compaction [91].

Transform coding methods can be interpreted as a rotation, or as a matrix manipulation. If $[f]$ denotes the matrix of the current image, $[T]$ the matrix of the transformation, and $[C]$ the transformed image. These three matrices are then linked by:

$$[C] = [T].[f].[T]^t \tag{4.1}$$

It is assumed that [T] is orthogonal, and the image can therefore be recovered:

$$[f] = [T]^t.[C].[T] \tag{4.2}$$

There exist several transform methods, such as Karhunen-Loeve Transform (KLT), or Discrete Cosine Transform (DCT). The KLT provides the best energy compaction but it requires the autocorrelation matrix of the current image and is therefore complex to implement. A 2D DCT is used instead most of the time , with 8×8 blocks (64 pixels) which provides a compromise between compression efficiency and block artifacts. For the 2D DCT, the coefficients of an $N \times N$ block of a transformed image $[C]$ are expressed as:

$$C(u,v) = \frac{2}{N} s(u)s(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) cos(\frac{(2i+1)u\pi}{2N}) cos(\frac{(2j+1)u\pi}{2N}) \tag{4.3}$$

$$s(x) = \frac{1}{\sqrt{2}} \; if \; x = 0, \; 1 \; otherwise \tag{4.4}$$

The coefficients of the inverse DCT of an $N \times N$ block are given by:

$$f(i,j) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} s(u)s(v)f(u,v) cos(\frac{(2i+1)u\pi}{2N}) cos(\frac{(2j+1)u\pi}{2N}) \tag{4.5}$$

- Quantisation: Due to the orthonormality of the previous transforms, there is equal energy in the pixel and the transform domains. Transform methods only provide energy compaction, but no energy compression. Quantisation and entropy coding do however provide energy compression. Quantisation exploits HVS characteristics. Human eyes are less sensitive to high frequencies which can therefore be quantised more heavily without affecting quality. The main idea of the quantisation is to force some coefficients to zero. There are several quantisation techniques. Details can be found in [90, 91].

  In video coding, the quantiser is often used as a regulator for the output bit rate. Decreasing the quantiser step size (or decreasing the quantisation parameter) will lead to sharper but more numerous output values, increasing therefore the bit rate. On the other hand, increasing the quantiser step size (or increasing the

quantisation parameter) will lead to less accurate value but less numerous output values, decreasing therefore the bit rate.

4.1.2 Temporal Redundancy Reduction

This refers to the inter frame coding. Parts of an image that are static, i.e. where there is no change from one frame to the following one, do not need to be coded. On the other hand, for parts with changes (movement, illumination variation), inter frame coding is used. The difference between the two adjacent pictures is coded, instead of each separate image independently. In addition, if the motion of an object can be estimated, the difference between the current image and the motion compensated image can be coded [91].

- Motion Estimation: This is one the main problems in video coding and comprises 60% of the complexity [91]. In most motion estimation algorithms, it is assumed that objects move in translation in a plane that is parallel to the camera plane (no zoom and no rotation), and it is assumed that illumination is spatially and temporally uniform. The most common algorithm is the Block Matching Motion Estimation (BMME) method and is depicted in figure 4.2.

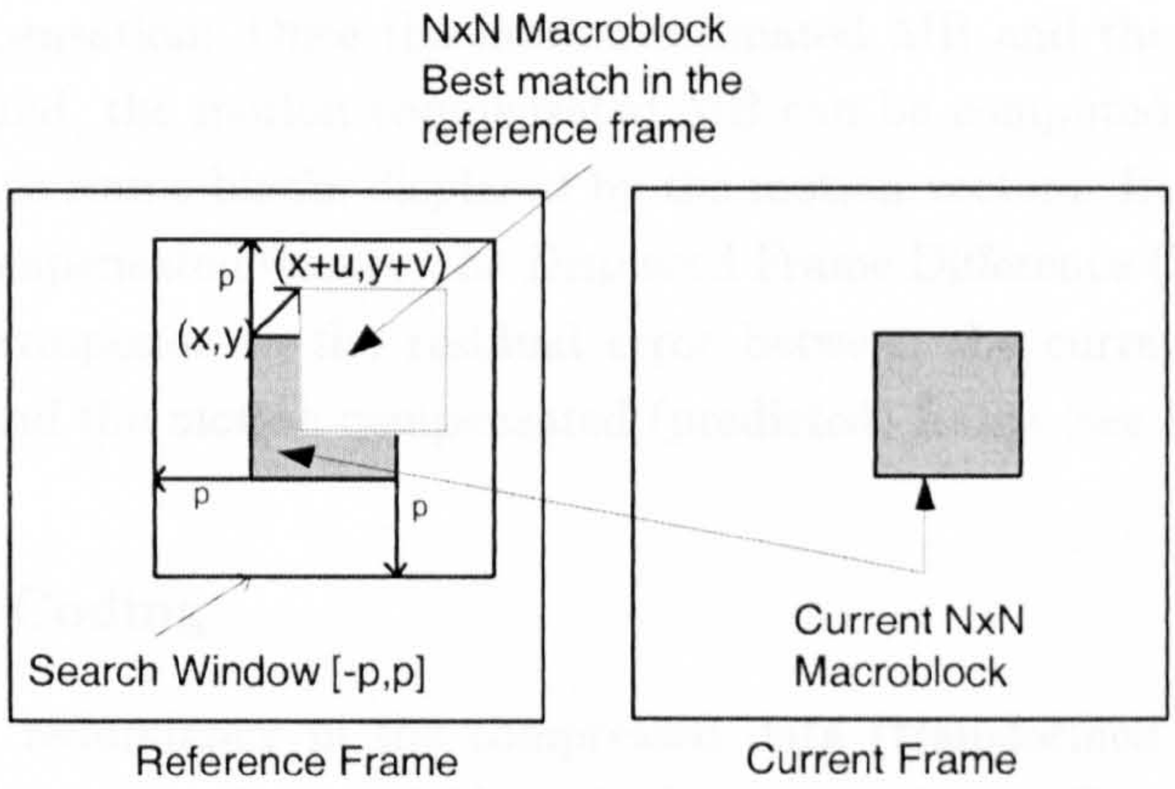


Figure 4.2: Block Matching Motion Estimation Principle

A frame is first divided into $N \times N$ blocks called macroblocks (MB), typically *N=16*. The aim is to determine the $N \times N$ MB in the reference picture that best matches the $N \times N$ MB in the current picture. The smaller N, the better the prediction. Practically, the reference picture does not need to be adjacent to the current frame. As it is impractical and unnecessary to search into the whole picture, a search region in the reference picture is required. The search is performed into a search window of size $[-p, p]$ defining the area in which to find the best match [90, 91, 92]. Assume the top left corner of the current MB to be coded has the coordinate (x, y) in the current picture. Assume the top left corner of the best match has the coordinate $(x + u, y + v)$ in the reference picture. The vector (u, v) is called the motion vector (MV) and is coded.

The matching is based on a criterion. The two most commonly used are the Mean Square Error (MSE) and the Mean Absolute Difference (MAD)and are defined as:

$$MSE(i,j) = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}[f(x+k,y+l) - g(x+k+i,y+l+j]^2 \qquad (4.6)$$

$$MAD(i,j) = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}|f(x+k,y+l) - g(x+k+i,y+l+j| \qquad (4.7)$$

where $-p \leq (i,j) \leq p$ is the coordinate of the searching point in the search window in the reference frame, $f(x+k,y+l)$ represents the pixel of the current block to be coded and $g(x+k+i,y+l+j)$ represents the pixel in the reference frame. To reduce the complexity, MAD is preferred to MSE. Fast Motion Estimation algorithms are used to reduce further the complexity. Note that the search can be refined by using sub-pixel accuracy. The motion estimation can be backward, forward or in both directions by using combinations [90, 91, 92].

- Motion Compensation: Once the motion estimated MB and the motion vector have been found, the motion compensated MB can be computed. This is equal to the reference frame blocks displaced by the motion vectors. Instead of coding the motion compensated frames, the Displaced Frame Difference (DFD) is coded. The DFD corresponds to the residual error between the current frame to be transmitted and the motion compensated (predicted) frame (see figure 4.3)

4.1.3 Entropy Coding

In order to reduce redundancy in the compressed data (transformed and quantised DFD and MVs), entropy coding and Variable Length Coding (VLC) is used. This allows short codewords to be assigned to the most probable input values, and long codewords to the less probable values. The length of the codewords varies inversely with the probability of occurrence of the values. The entropy of the symbols (input values) can be viewed as the minimum average number of bits required to code them and can be expressed as:

$$H(x) = -\sum_{i=1}^{n} p_i log_2 p_i \qquad (4.8)$$

where p_i is the probability of the i^{th} symbol.

In video coding, the two main VLC coding methods are Huffman coding and Arithmetic coding. Huffman coding generates a prefix code, i.e. each codeword can not be a prefix of another codeword. Arithmetic coding can achieve the theoretical entropy bound. Details on Huffman and Arithmetic coding can be found in [90, 91, 92, 93].

4.1.4 Video Codec Block Diagram

Since video coding uses both spatial (DCT+Q) and temporal redundancy (Motion Estimation) reduction techniques, the technique is denominated as hybrid. Figure 4.3 shows the block diagrams of the hybrid encoder and the decoder [90, 91, 92, 94].

Every MB is divided into four 8×8 luminance (Y) blocks. Depending the sub-sampling ratio of the video format, the number of chrominance (U and V) blocks per MB varies from one to four. Then each block is DCT coded and quantised. The motion estimator requires the current frame and the reference frame (which is the previous reconstructed frame) and generates the motion vector. The motion estimation is generally carried out only on the luminance components. The motion compensator requires the reference frame and the motion vector and generates the predicted frame.

4.1.5 Video Quality Assessment

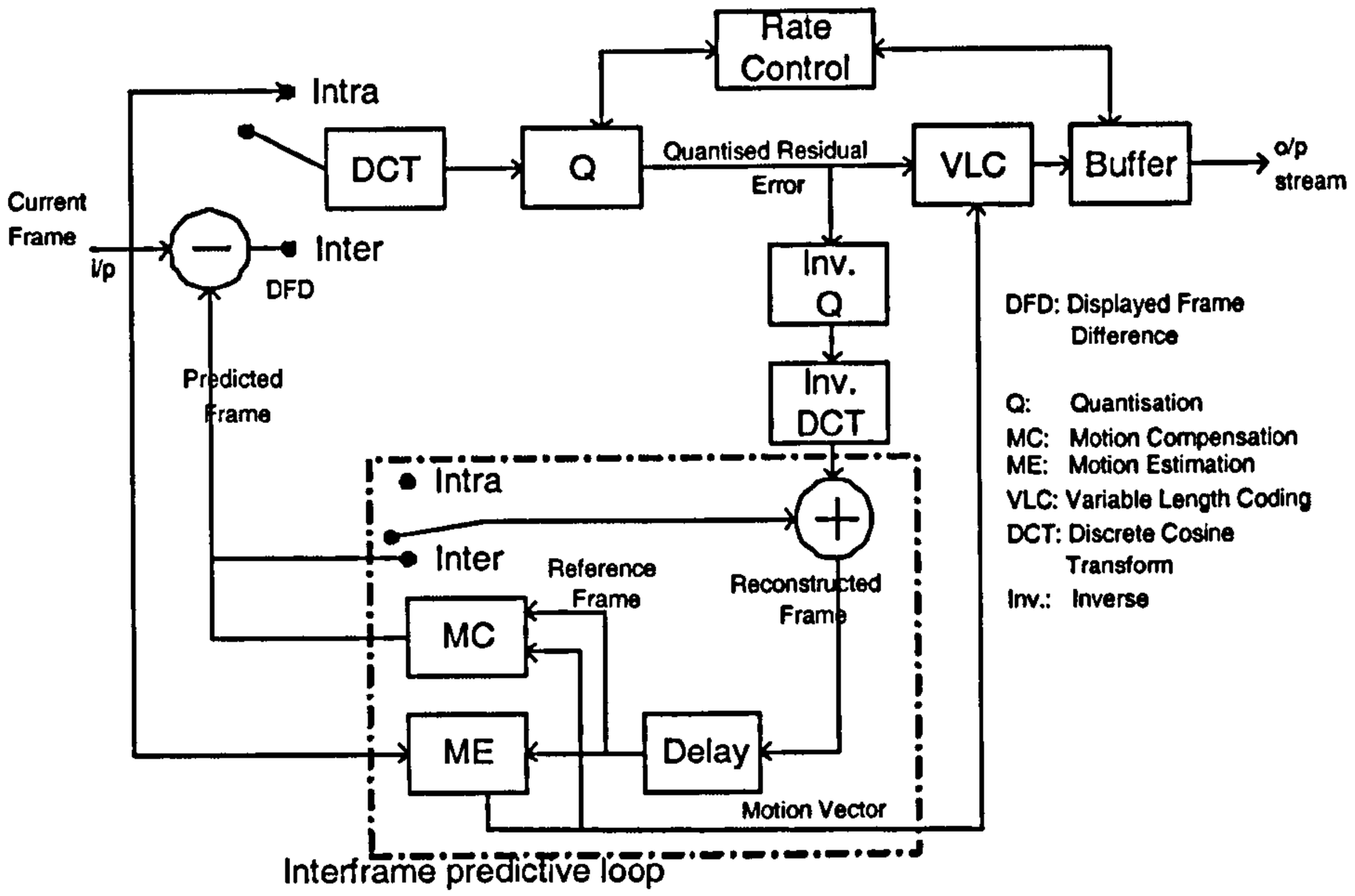
Picture quality has been traditionally subjectively assessed, i.e. pictures are shown to a group of subjects, and their views on the perceived quality of distortions are sought [91]. Distortions may not be noticeable for an observer, especially with a good coding scheme, and they may differ from one observer to another [90]. Subjective measurements are however supplemented by objective measurements and results collected from the different persons in the group are then processed with different algorithms depending on the assessment problems: comparison between systems, measure of quality. The various subjective assessments techniques for the quality of television pictures can be found in [95].

The Peak-Signal-to-Noise Ratio (PSNR) is one of the commonest way to assess an image. It is based on the Mean-Square-Error (MSE).

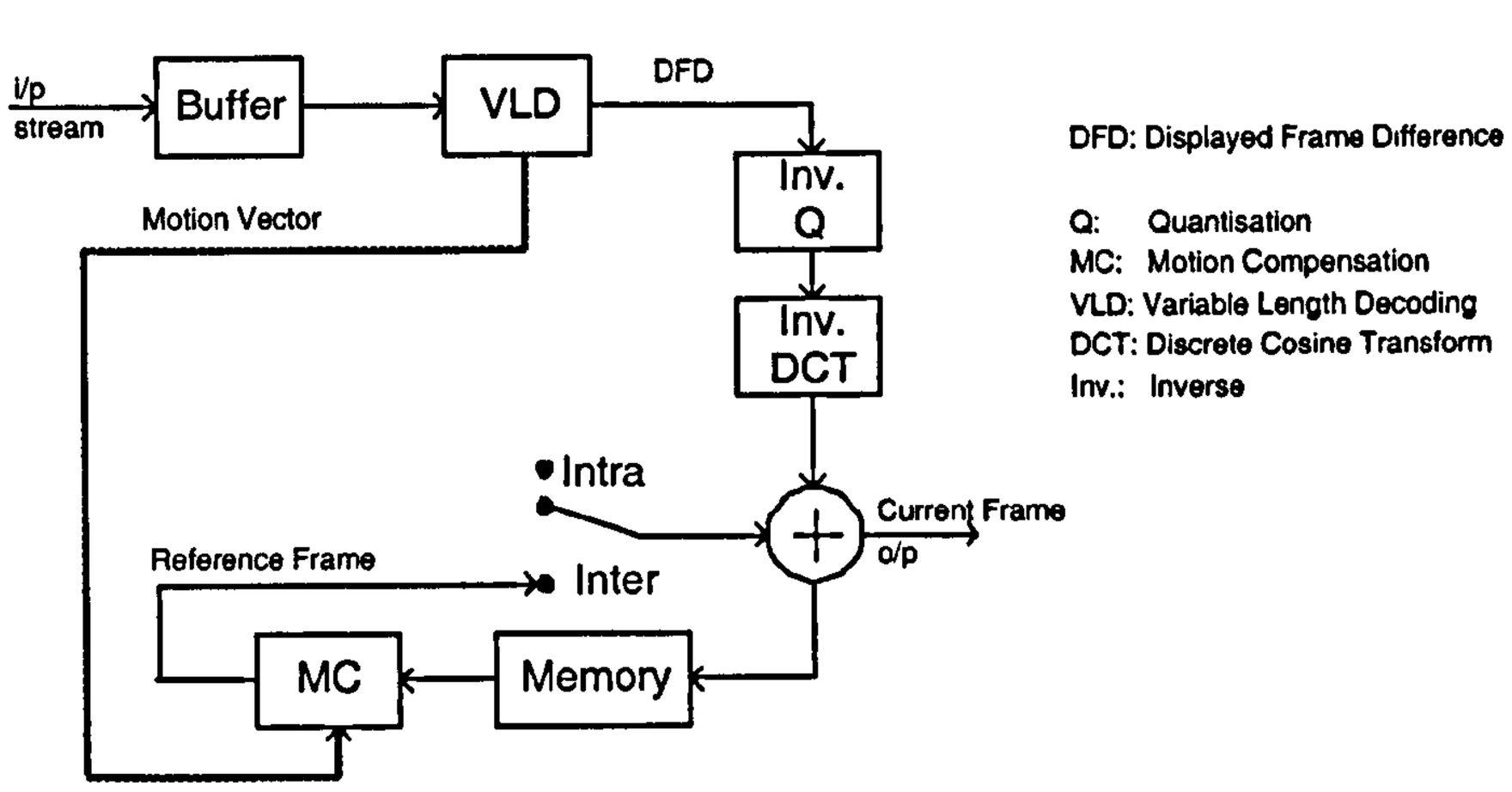
$$PSNR = 10log_{10}\left(\frac{255^2}{MSE}\right) \tag{4.9}$$

$$MSE = \frac{1}{N \times M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[x_{reference}(i,j) - x_{current}(i,j)]^2 \tag{4.10}$$

where $x_{reference}$ represents a pixel of the reference image (e.g. original), $x_{current}$ represents the pixel of the current image (e.g. coded) and M and N are the width and height of the picture in pixels.



(a) Coder



(b) Decoder

Figure 4.3: Generic hybrid Video Codec Block Diagram

4.2 MPEG-2

Full details on the coding techniques implemented in the MPEG-2 standard can be found in specific journal publications, conference proceedings and video coding books among which [90, 91, 92, 94, 96] are recommended. Key points are however highlighted in this section.

4.2.1 MPEG-2 Coding Layer

4.2.1.1 Coding Features

Following the development of the MPEG-1 standard [97] in the early 1990s, mainly aimed at storage, the ISO/IEC developed the MPEG-2 standard [98], also known as the H.262 standard. It allows higher bit rates than MPEG-1 (from 2 to 30Mbits/s) and has been designed to suit a wider range of applications such as Digital TV, Satellite broadcasting, Cable TV, High Definition TV (HDTV), DVD and Interactive Storage Media. MPEG-2 uses a similar hybrid block diagram to figure 4.3 and uses many of the MPEG-1 features [96]. Among them are the following:

- A zig-zag scanning of the quantised components prior to the VLC (see figure 4.4) is implemented in order to increase the efficiency of capturing non-zero components.

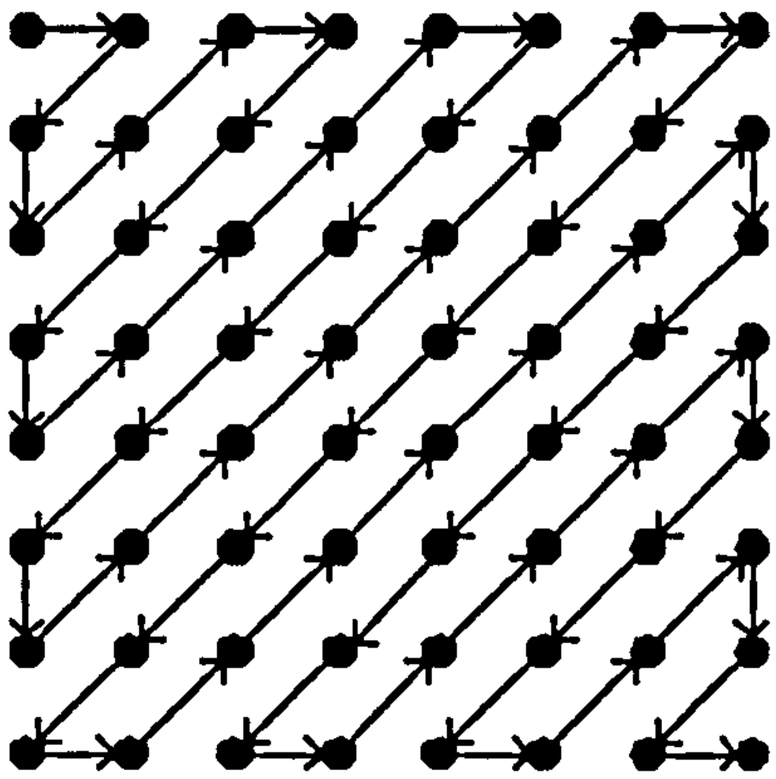


Figure 4.4: Zig Zag Scanning

- The 2D-VLC coding is performed on a *run/index* basis. *Run* corresponds to the number of zeros preceding the value to be coded, and *index* corresponds to the index of the magnitude of the value to be coded.
- There are three main picture types:
  - Intra (I) frames are coded without reference to previous frames and are intra-coded. They provide therefore an access point to the coded sequence.
  - Predicted (P) frames are predictively coded with reference to either a previous I or P frame

- Bi-Predicted (B) frames are bi-directionally predicted. They may use, past future or a combination of both picture for prediction.

- A layered structure is used: video sequence layer, Group of Pictures (GoP) layer, picture layer, slice layer, MB layer and finally the block layer. A GoP is composed of a series of one or more pictures, with an I picture always being the first, it is characterised by its length N and by the length M between the I frame and the first P frame. Figure 4.5 shows an example of GoP with N=*9* and M=*3*. Note that the coding order differs from the display order since the coding of frames B_2 and B_3 require future frame P_4 to be coded first. A slice is a group of consecutive MBs and is used in order to prevent channel error propagation. In theory, they can have different sizes within a picture and may start and end at any MB of the picture. However, for MPEG-2, a slice is restricted to one row.

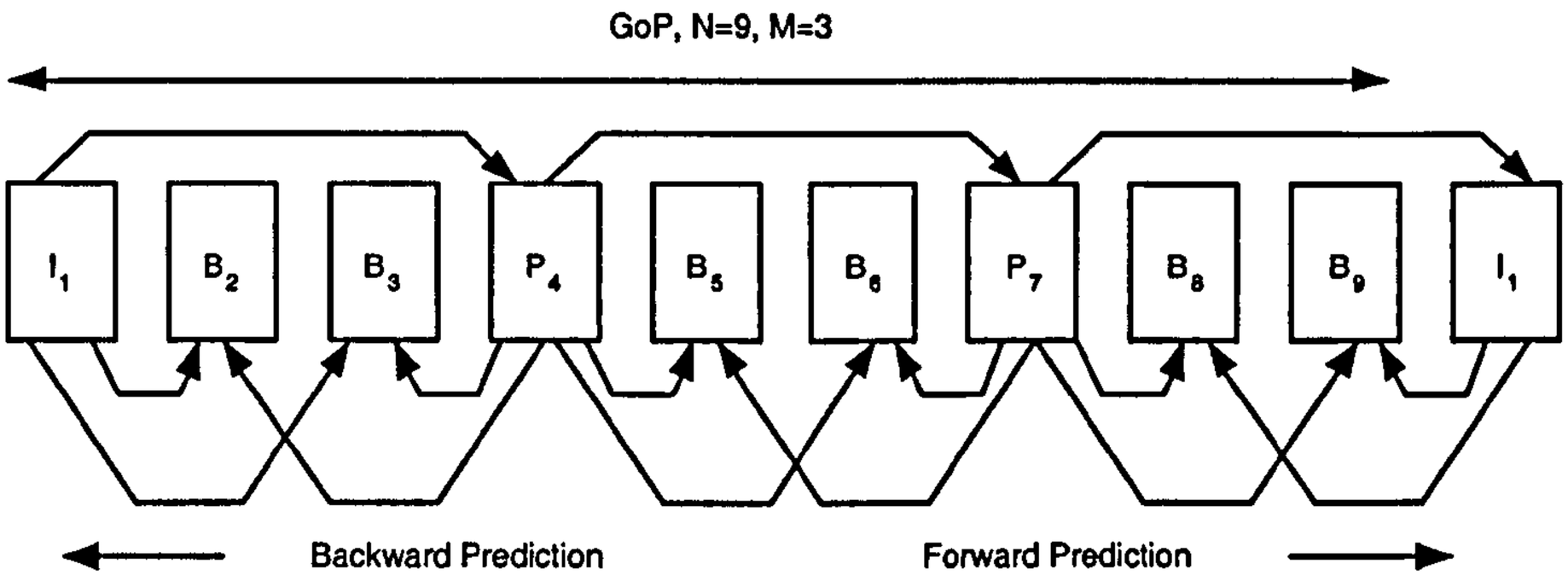


Figure 4.5: Group of Pictures

- The motion estimation used is bi-directional, and is done at half-pixel resolution.It is carried on a MB basis (16×16). The DCT is performed at the block (8×8) level. The control of bit rate is achieved by tuning the quantisation process.

MPEG-2 introduces new features such as larger picture resolution ranges, support for interlaced sequences, motion estimation at a 16x8 level for interlaced pictures and scalability (temporal, spatial and SNR). Because of interlaced pictures, an alternate scanning is also allowed. A linear and a non-linear quantisation process are supported to increases the precision at high bit rates.

4.2.1.2 Bit Stream Syntax

The MPEG-2 bit stream syntax follows the layered structure of the coding using a hierarchy with headers. The bit stream uses *Start-Codes* to define the next field in the bit stream. Each Start-Code is composed of 32 bits and consists of a Prefix Start-Code (Psc) followed by a Start-Code ID. The Psc is a string of 23 or more binary zeros followed by a binary one. The Start-Code ID consists of two hexadecimal characters that identify the type of Start-Code. A Start-code is always followed by the header

of the corresponding type, which is either Sequence, Video, GoP, Picture, or Slice header. Note that there is no Start-Code before a MB header. Start Codes are used for resynchronisation.

4.2.1.3 Profiles and Levels

One of the main philosophies of the MPEG-2 video standard is that it has to be 'generic', meaning that the coding is not targeted for a specific application. The standard includes many algorithms/tools that can be used for a variety of applications under different operating conditions. Because of the numerous algorithms and tools supported, MPEG-2 defines *profiles* (subsets of the bit stream syntax to be addressed to specific classes of applications) and profiles are partitioned into *levels* (set of constraints imposed on parameters in the bit streams such as frame size, resolution). Levels and Profiles have hierarchical relationships: higher profiles also support lower profiles, and higher levels support lower levels.

4.2.1.4 Scalability

In order to support specific applications, bit streams can be ordered into layers. When two or more layers are generated, the coded video data are called scalable video bit streams. The first layer, called the base layer, is coded independently and contains the basic data which will exhibit a reasonable quality when decoded on its own. The other layers are called enhancement layers and contain data that refine the base layer [91, 96]. Scalable video coding is often referred as layered coding or hierarchical video coding. One of the aims of scalable video coding is to increase robustness of video codecs against packet loss, especially during transmission over noisy channels. It can then provide error resilience [91, 96]. Decoders are able to extract a hierarchical structure from a single layered coding bit stream and select the layers their applications can use.

MPEG-2 defines four basic and one hybrid scalable modes:

- *Data Partitioning* is the simplest scalable tool of MPEG-2. This tool divides the bit stream of a single layer non scalable MPEG-2 into two partitions. The first layer comprise the critical parts, such as headers, MVs and lower order DCT coefficients, whereas the second layer comprises less critical data, such as higher DCT coefficients. This technique offers however a sensitive base layer with poor quality [96].

- *SNR Scalability* is used in applications involving telecommunication and multiple quality video services. The two generated layers have the same spatio-temporal resolution but with different qualities. The basic layer is coded by itself to provide the basic video quality at a low bit rate. The difference between the reconstructed base layer and the input video is coded by a second encoder with a higher precision to generate the enhancement layer. The enhancement layer, when added back to the base layer, regenerates a higher reproduction of the input video. The two bit streams are then multiplexed together [96].

- *Spatial Scalability* generates two spatial resolution video streams from a single source. The base layer is coded by itself with the basic spectral resolution, and the enhancement layer carries full spatial resolution. The base layer can use SIF or lower picture resolution at 4:2:0, 4:2:2 or 4:1:1 formats, while the enhancement layer uses ITU-R-601 format with 4:2:2. The base layer is generated by down-sampling spatially the input sequence. The base layer is decoded and then up-sampled, the difference with input sequence is then coded generating the enhancement layer [96].

- *Temporal Scalability* partitions the video sequence into two layers in which the base layer is coded to provide basic temporal rate and in which the enhancement layer is coded with temporal prediction with respect to the base layer. A simple switch may achieve this method. For example I and P frames can be in the base layer, whereas the B frames are in the enhancement layer [96].

- *Hybrid Scalability* is a combination of individual scalabilities (SNR, spatial or temporal). If two scalabilities are combined, then three layers are generated [96].

4.2.2 MPEG-2 System Layer

The system section of the MPEG-2 standard [99] specifies how the compressed video, audio and data elementary streams may be packetised and multiplexed together to form a single data stream [91, 96, 100]. Elementary streams can be multiplexed into program streams or transport streams. MPEG-2 enables the delivery of multiple programs simultaneously within a single transport stream, without requiring them to have a common time base. This section provides a brief review of MPEG-2 elementary streams (ES), packet elementary streams (PES), program streams (PS) and transport streams (TS), including the full description of the PES header and TS header.

A program in MPEG-2 is defined as a single broadcast service and is composed of one or more elementary streams. An elementary stream is a single, digitally coded (and MPEG-2 compressed) component of a program. There are four types of elementary streams: audio, video, systems data, and other data (such as user data or teletext information) [100]. Figure 4.6 shows how these elementary streams are multiplexed.

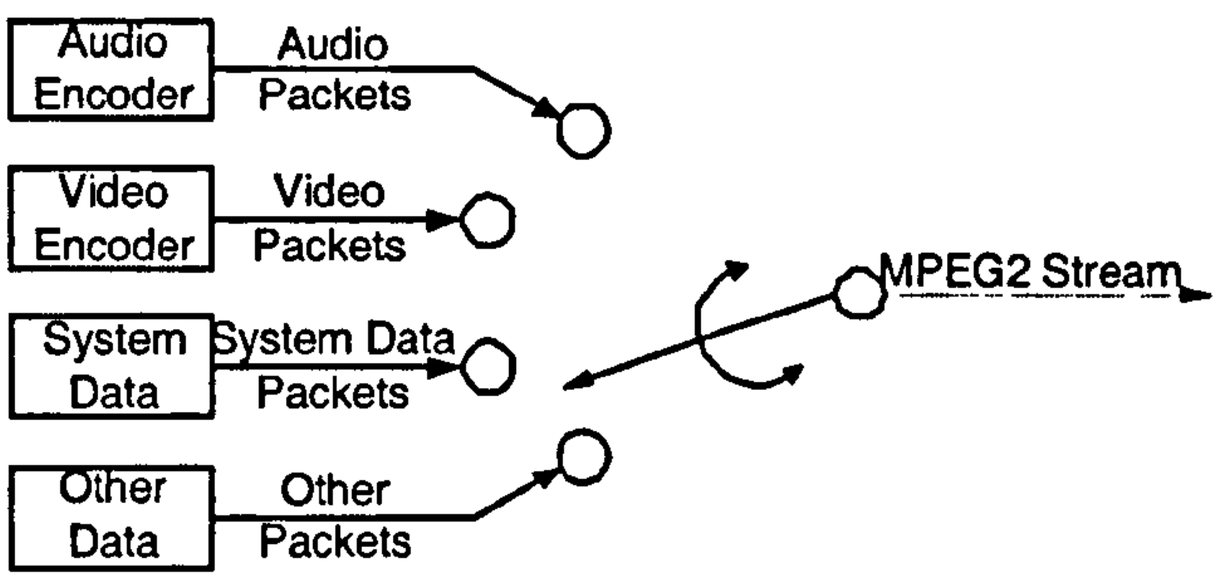


Figure 4.6: Elementary stream multiplexing

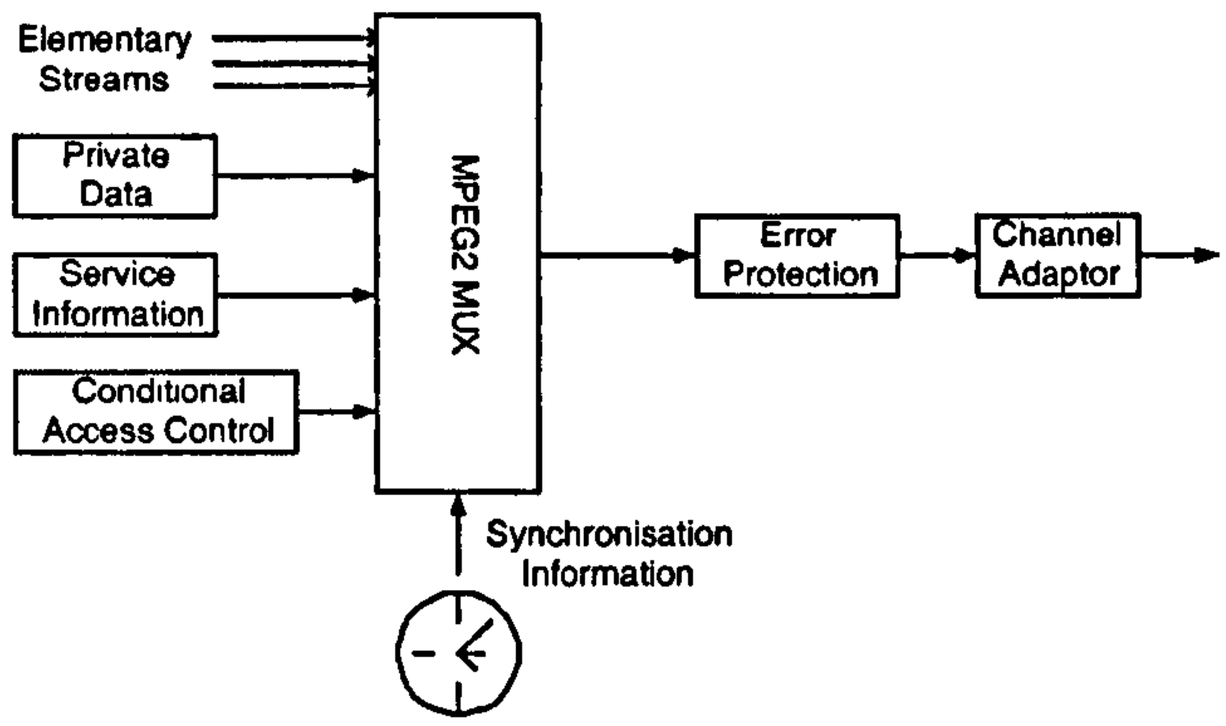


Figure 4.7: MPEG-2 multiplexer

The output of the MPEG-2 multiplexer is a contiguous stream of data bytes. The various ESs that have been multiplexed need to be synchronised with each other. Figure 4.7 shows the whole MPEG-2 multiplexer including synchronisation [100]. A system of time stamps is specified to ensure that related ESs are replayed in synchronism at the decoder. Service information may include network parameters and information about the ESs and support is provided for the control of scrambling, for conditional access. There is no error protection within the multiplex; this and the subsequent modulation (channel adaptation) are chosen to suit the channel (or storage medium). The MPEG-2 systems specification defines two alternative multiplexes: the programme stream and the transport stream. Each is optimised for a different set of applications.

4.2.2.1 Packet Elementary Stream

A decoded MPEG-2 picture is known as a presentation unit and an MPEG-2 coded representation of a picture (or part of a picture) is called a video access unit. A video elementary stream consists of a succession of variable length video access units. Similar terminology is used for audio. The packetiser converts each elementary stream into a packetised elementary stream (PES), which is limited to 64 kbytes and consists of a variable length PES packet header and variable length PES packet payload. PES payloads contain data bytes taken sequentially from the original ES. Even if there is no requirement to align the start of access units and the start of PES packet payload, it is recommended that PES packetisation should be performed at the access unit boundaries (frame or slice) so that it is aligned with resynchronisation start-codes. The PES packet header is shown in figure 4.8 and is composed of:

- A so-called PES packet start code prefix (PSC) comprising 3 bytes equal to 0x000001 (two bytes equal to 0 and one byte equal to 1).

- The stream ID (1 byte) serving to label the stream as well as to specify the type of stream and distinguish PES packets belonging to one ES from those of another ES within the same program.

- The PES packet length (2 bytes) specifying the number of bytes that follows in the PES packet (including the remaining bytes of the header).
- Flag 1 and flag 2 (1 byte each) containing indicator bits which show the presence or absence of the various optional fields that may be included in the PES header. Among them the following are noteworthy:
  - P bit and D bit indicating the presence of presentation time stamps (PTS) and decoding time stamps (DTS) fields within the PES header.
  - CRC flag, Scrambling Flag, Priority Flag, Copyright flag.
  - PES extension Flag indicating the presence of extension in the PES header.
- The PES header length (1 byte) specifying the total number of bytes following in the PES optional field before the first byte of PES payload is reached.

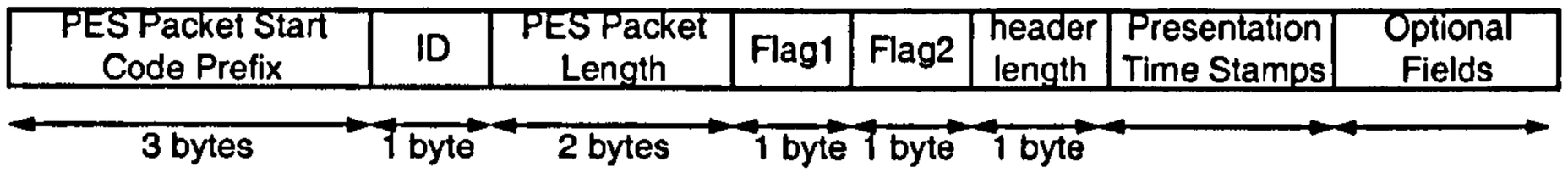


Figure 4.8: PES packet header

Figure 4.9 shows the MPEG-2 systems multiplex from PES to the two streams available: programme and transport streams [91].

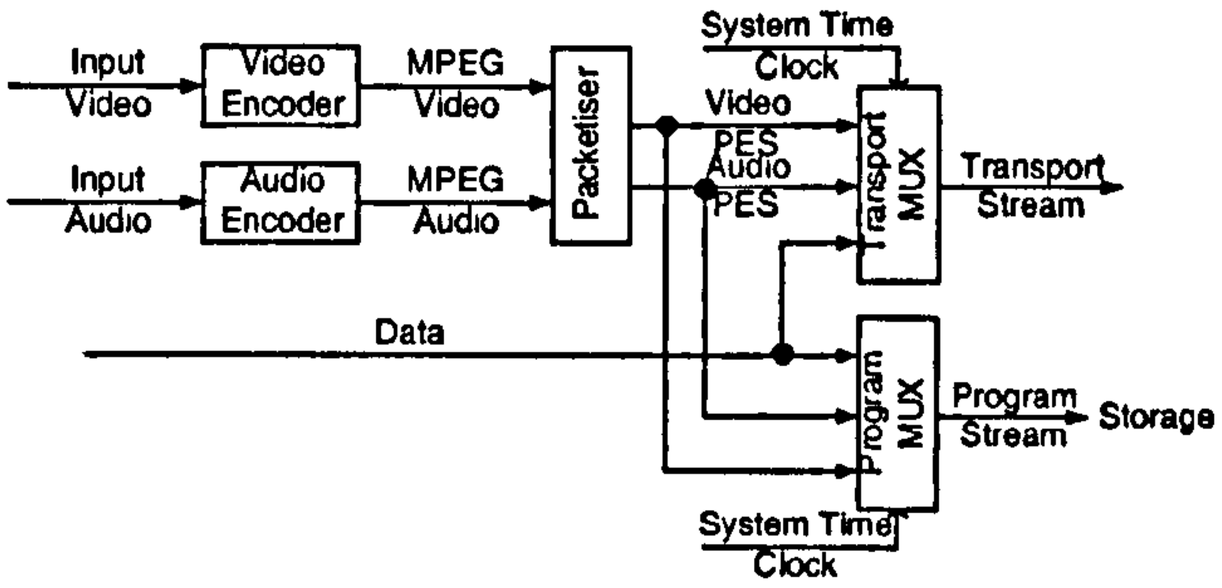


Figure 4.9: MPEG-2 systems multiplex of PS and TS

4.2.2.2 Program Stream

A program stream is based on the MPEG-1 [97] multiplex and is compatible with MPEG-1. It is intended for storage and error-free retrieval of program material from digital storage media and lacks the error control mechanisms which are present in the transport stream. A program stream comprises a succession of variable-length and multiplexed PES packets (with a maximum length of 64 kbytes). Each packet begins with a header, the corruption of which will cause the loss of the entire packet. If the packet length field is corrupted, subsequent packets may also be corrupted [100].

4.2.2.3 Transport Stream

The transport stream of MPEG-2 is intended for use in error-prone transmission environments and it is therefore suitably protected. It is composed of a succession of 188-byte packets (transport stream packets), each of which is provided with error protection (such as Reed-Solomon). The first byte of each TS packet is a synchronisation byte and the remainder may contain multiple video, audio and data streams. This is a preferred method for conveying multimedia streams, but is more complex to create and multiplex than a PS [100, 101]. PES packets are packetised into TS packets for transmission as shown in figure 4.10.

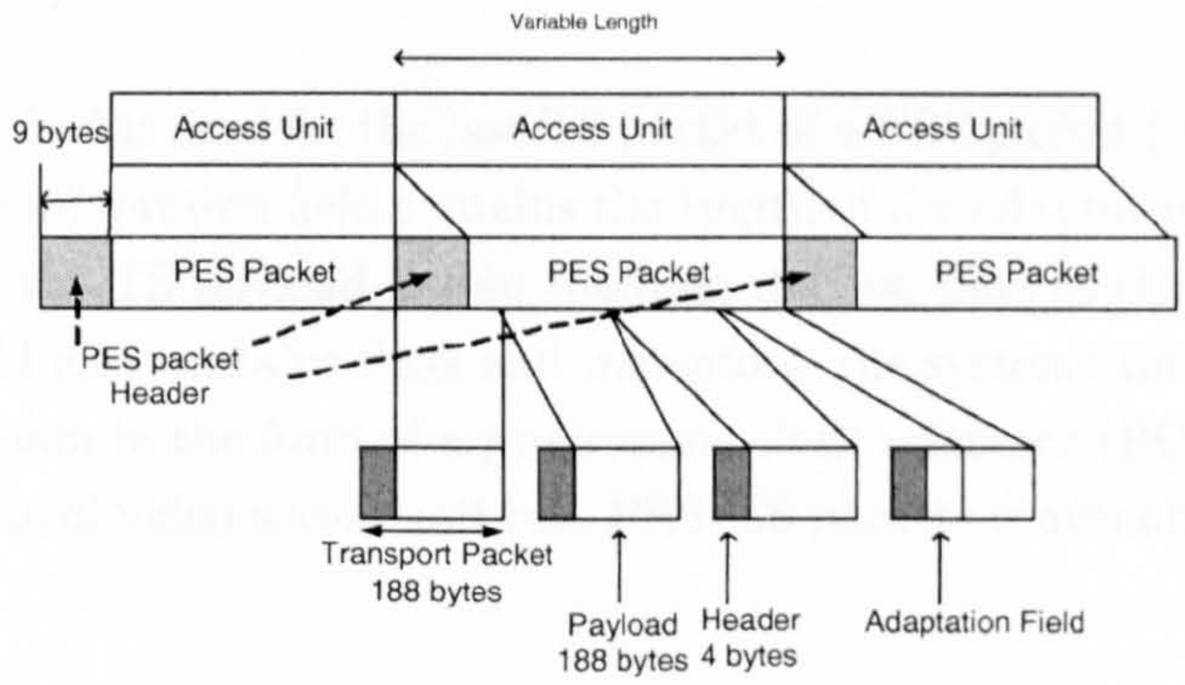


Figure 4.10: Generation of TS packet from the video bit stream

A TS packet is composed of a 4-byte header (figure 4.11) and a 184-byte payload taken sequentially from the PES-packet, except for the last TS packet which contains an adaptation field. The TS packet header is composed of:

- Synchronisation byte with a fixed value of 0x47.
- Transport error indicator (tei) of 1 bit indicating that an uncorrectable bit-error exists in the current TS packet. This indicator is raised by underlying networks.
- Payload unit start indicator (pus_id) of 1 bit indicating the presence of a new PES packet in the TS payload.
- Transport priority (tp) of 1 bit indicating a higher priority than other packets.
- Packet identifier (pid) of 13 bits used to distinguish TS packet containing data from one ES from those carrying data of other ESs.
- Transport scrambling control (tsc) of 2 bits indicating the scrambling mode of the packet payload.
- Adaptation field control (afc) of 2 bits indicating the presence of the an adaptation field or payload.
- Continuity count (cc) of 4 bits incremented between successive TS packets having the same pid.

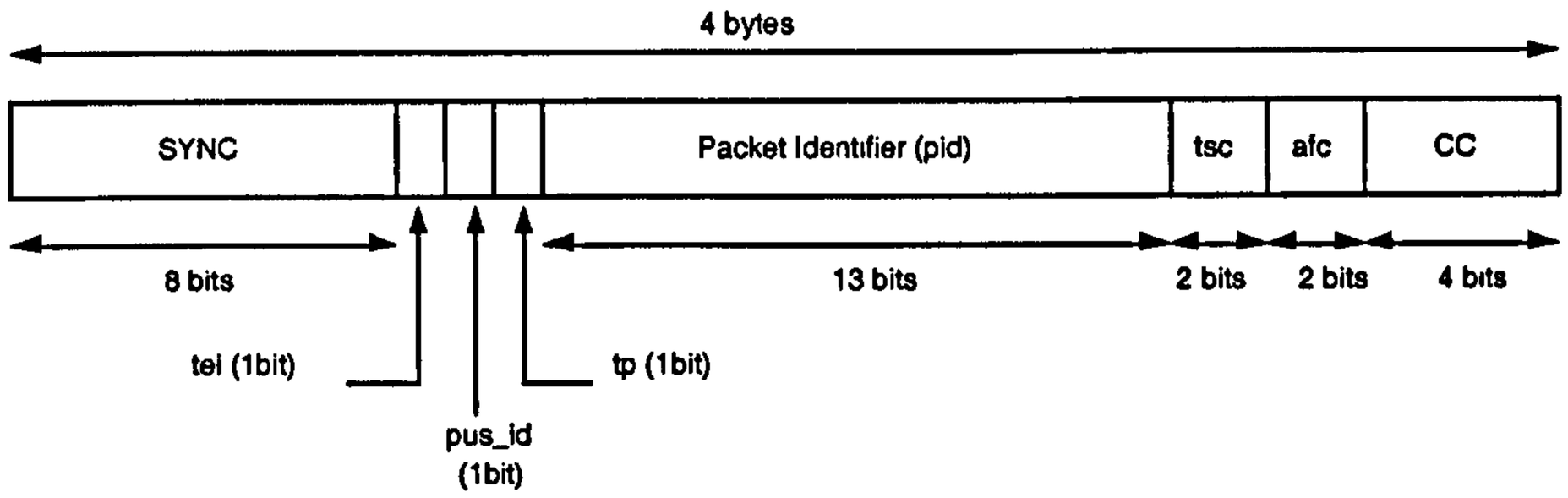


Figure 4.11: TS packet header

The adaptation field is used for the last TS packet of a PES packet (see figure 4.10). The first byte of the adaptation field contains the length of the adaptation field, i.e. the number of bytes in the TS payload before reaching the last part of the PES payload. The adaptation field also contains flags and indicators, the systems time clock (STC) and timing information in the form of a programme clock reference (PCR). A detailed study of the mapping of video access unit into PES/TS packets is available in appendix E.

4.3 Error Resilience and Concealment Techniques

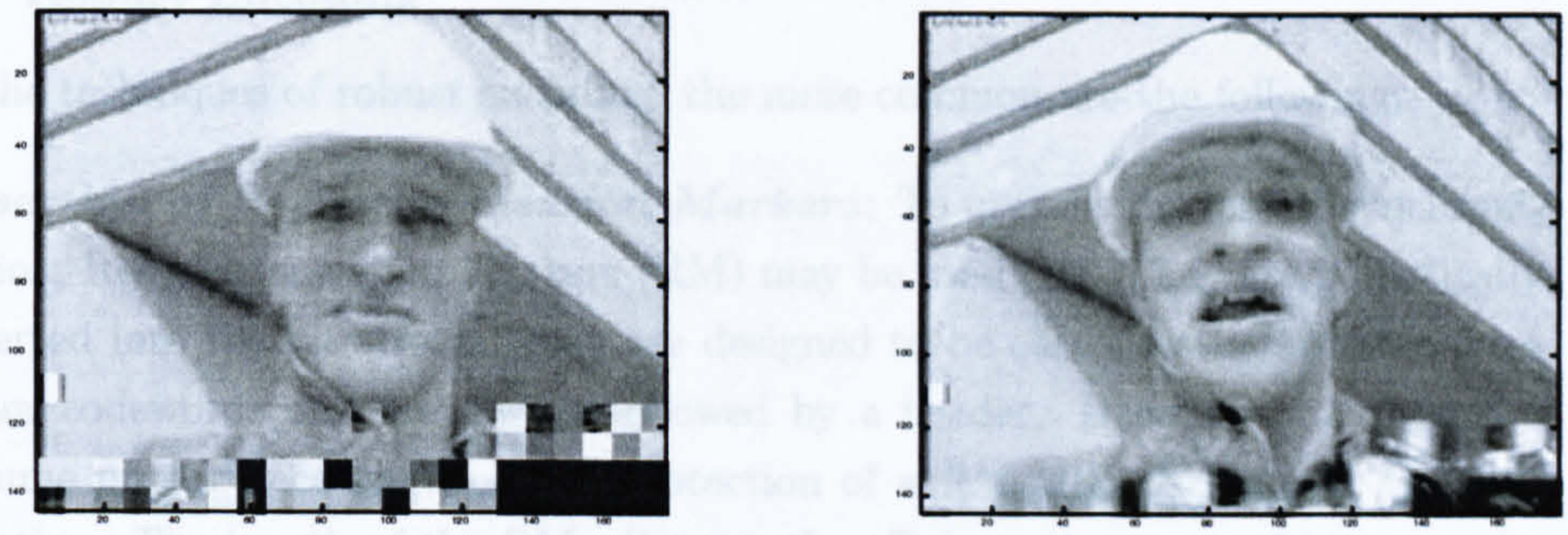
Since video compression uses predictive coding, temporal prediction and VLC coding, encoded video is very sensitive to transmission errors. Error control in video transmission is therefore very challenging [102].

4.3.1 Detection and Effects of Errors

The detection of an error is performed at the receiver side. It can be done with the addition of header information (e.g. sequence number in packet switch network), check sum or Forward Error Coding (FEC) at the transport coder/decoder. Error detection can be done at the video level (video decoder) by using the difference of pixel values between neighbouring lines with Differential Pulse Code Modulation (DPCM) [102]. The best error detection method is FEC with the addition of header information [11]. Two kinds of errors can be highlighted: random bits are inverted, inserted or deleted due to imperfections of the physical channel. These imperfections can desynchronise the VLC coded information. Decoders may be able to detect errors when trying to decode an unrecognisable codeword. The other type of error is the erasure error due to packet loss, burst errors or system failure [11]. This second type is much more destructive than random bit errors.

Because of the use of predictive coding, an error may propagate to adjacent MBs (spatial propagation) and to adjacent frames (temporal propagation). Because of the use of variable length coding, an error will either make the decoder decode the wrong codeword or make a codeword unrecognisable. The latter case will lead to the loss

of synchronisation where all the following bits are discarded until resynchronisation is re-gained, even if they are correctly received. Figure 4.12 shows how errors propagate temporally. Frame 0 is received corrupted. All the upcoming frames are received error free. However, because of the temporal prediction, errors are still present in frame 20.



(a) Frame 0 Received Corrupted

(b) Frame 20 Received Error Free

Figure 4.12: Effect of Error

Figure 4.3.1 shows how the PSNR of one video sequence evolves in the presence of errors. The sequence is received error free until frame 42 is corrupted. Because of the error, the PSNR drops. Because of the intra coding of some MBs that cannot propagate the error, the PSNR slowly recovers until another error occurs, dropping the PSNR again.

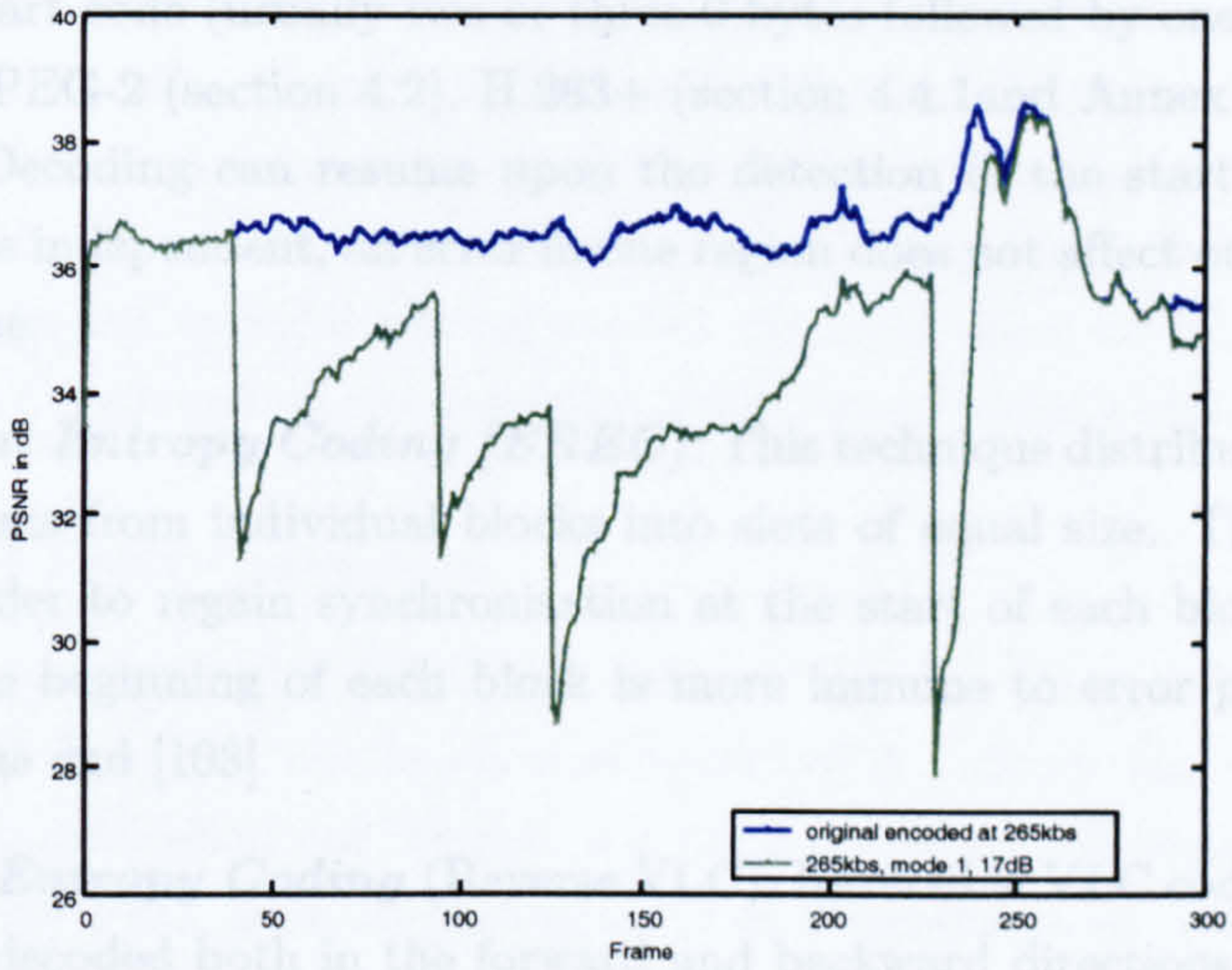


Figure 4.13: Error Propagation with H.263+ Video Standards with no I frame

4.3.2 Forward Error Concealment: Error Resilience Coding

In Forward Error Concealment, the source coder has the primary role. This is to make the bit stream more resilient to potential errors and to ensure that, if an error occurs,

it will not have a disastrous effect. Most of the time, forward error resilience techniques increase the bit rate by adding redundancy [102]. The goal is to have the maximum gain in error resilience with the smallest amount of redundancy, either by helping to prevent errors or by enabling the decoder to perform better concealment [11] or by allowing a graceful degradation rather than a dramatic change in of the quality.

4.3.2.1 Robust Encoding

Among the techniques of robust encoding, the more common are the following:

1. ***Insertion of Resynchronisation Markers***: To prevent the loss of synchronisation, Resynchronisation Markers (RM) may be used [11]. They are periodically inserted into the bit stream, they are designed to be easily distinguishable from other codewords and are always followed by a header. Decoders can therefore resume proper decoding upon the detection of a RM after the loss os synchronisation. The length of the RMs dictates the efficiency in terms of bit rate, the periodicity of RMs shows how robust the bit stream is and how big the affected areas are. When an error occurs, the decoder discards all the bits from the error until the next RM.

2. ***Insertion of Intra-Block or Intra Frame***: The insertion of intra blocks or intra frame stops the temporal propagation of errors.

3. ***Independent Region Prediction***: Another way of dealing with loss of synchronisation is to use a partitioning of the frame into regions such as slices (sequence of adjacent MBs) or group of blocks (GoB) (one row of MBs). These entities are independent and are recognisable in the bit stream with their header always prefixed by a start code (usually two or three 0 bytes followed by one byte equal to one, as in MPEG-2 (section 4.2), H.263+ (section 4.4.1and Annex B of H.264 (section 4.4). Decoding can resume upon the detection of the start codes. As these regions are independent, an error in one region does not affect other regions within one frame.

4. ***Error Resilient Entropy Coding (EREC)***: This technique distributes variable-length bit streams from individual blocks into slots of equal size. This method allows the decoder to regain synchronisation at the start of each block. It also ensures that the beginning of each block is more immune to error propagation than those at the end [103].

5. ***Bidirectional Entropy Coding*** (Reverse VLC): Reversible VLC codewords are designed to be decoded both in the forward and backward directions. When an error occurs, the decoder discards all the bits until the next resynchronisation codeword where the decoder usually resumes the decoding process [1, 11]. Discarded bits may however be correctly received but could not be decoded because of the loss of synchronisation. Using Reverse VLC allows decoding in the reverse direction and recover discard and correct bits.

4.3.2.2 Layered Coding with Unequal Protection

This technique requires the bit stream to be partitioned into:

- one base layer containing the essential information that can be used to generate output with a low but acceptable quality
- enhancement layers that contain refinement information to obtain higher quality.

To combat channel errors, unequal protection (or transport prioritisation) will be done. The base layer is delivered with a higher degree of error protection. This can be done by different means: more reliable channel, stronger FEC, ARQ [11]. This ensures to receive correctly the base layer, whereas enhancement layers do no need to be protected that strongly. Since they only improve quality, errors in enhancement layers are less important [102].

There are several ways to obtain base and enhancement layers. Subband coding with up-sampling and down-sampling provides different levels (layers) of quality. MPEG-2 provides several scalability schemes, such as SNR, temporal and spatial scalability that generate base and enhancement layers (see section 4.2.1.4).

4.3.2.3 Multiple Description Coding (MDC)

With this technique, video data are first divided into several groups (descriptions) and are transmitted over separated channels. These channels are assumed to be independent. The probability that all channels experience losses at the same time is then small [102]. These can be different physical paths, or frequency division. Descriptions are correlated to each other, and have the same importance. They also must carry sufficient information to be acceptably decoded alone, and there is then overlapped information between several descriptions. Quality is improved with more descriptions. It therefore reduces coding efficiency compared to a single description coding, but it increases error robustness to long burst and/or channel failure [11]. Descriptions can be obtained by simply splitting adjacent samples among several channels using an interleaving sub-sampling lattice and then coding the resulting sub-images independently.

4.3.2.4 Forward Error Correction

Forward Error Correcting codes (FEC) are mainly used for error detection and for error correction. It increases overheads and therefore it reduces the bandwidth for data. Since video transmission can tolerate a certain degree of loss, some studies have been performed with FEC, such as combination of Reed-Solomon codes with block interleaving [102].

4.3.3 Error Concealment by Post-Processing

In this section it is assumed that the location of the errors is known. In Error Concealment, the source and transport decoders have the primary role. This role is to estimate

transmitted samples that are known to be erroneous or are not received. Error concealment techniques in contrast with error resilience coding do not increase the bit rate but add complexity to the decoder [102]. There are three types of data that need to be estimated in a damaged MB: texture information (i.e. pixel or DCT coefficient of the prediction error block), the motion vector (for P and B coding modes) and the coding modes [11].

4.3.3.1 Recovery of Texture

To recover the texture of the missing MBs, there exist many algorithms among which the following are highlighted.

1. ***Motion Compensated Temporal Prediction***: This technique is simple and effective. The texture of the corresponding MB in the previous decoded frame is used in the place of the damaged texture of the current MB. The MV remains the current MV (if undamaged). This method relies on the availability of the MV. If it is not available, it needs to be estimated as well [11].

2. ***Spatial Interpolation***: This technique is also simple. Pixels in a damaged block are interpolated from pixels in adjacent correctly received blocks. It is also possible to estimate the mean value (DC value) of a damaged block and replace it by a constant equal to the estimated DC value [11].

4.3.3.2 Recovery of Coding Mode and Motion Vector

Common solutions to recover the coding mode and the MV of the missing MBs are as follows.

1. ***Coding mode***: A simple approach to recover the coding mode is to assume that the MB is coded in intramode, and to use only spatial interpolation for recovering the underlying blocks. Another way is to collect the statistics of the coding mode pattern of adjacent MBs, and to find a most likely mode given the modes of surrounding MBs [11].

2. ***Motion Vector***: If the video has slow motion, the damaged (or lost) MV can be assumed to be zero. The MV of the corresponding block in the previous frame can be used. The average (or the median value) of all MVs from spatially adjacent blocks can also be used [11].

4.3.3.3 Previous Frame Copy

The Previous Frame Copy (PFC) algorithm simply replaces the missing MBs by the MBs spatially positioned at the same place but from the previous frame. This technique is not possible for the first frame of the sequence.

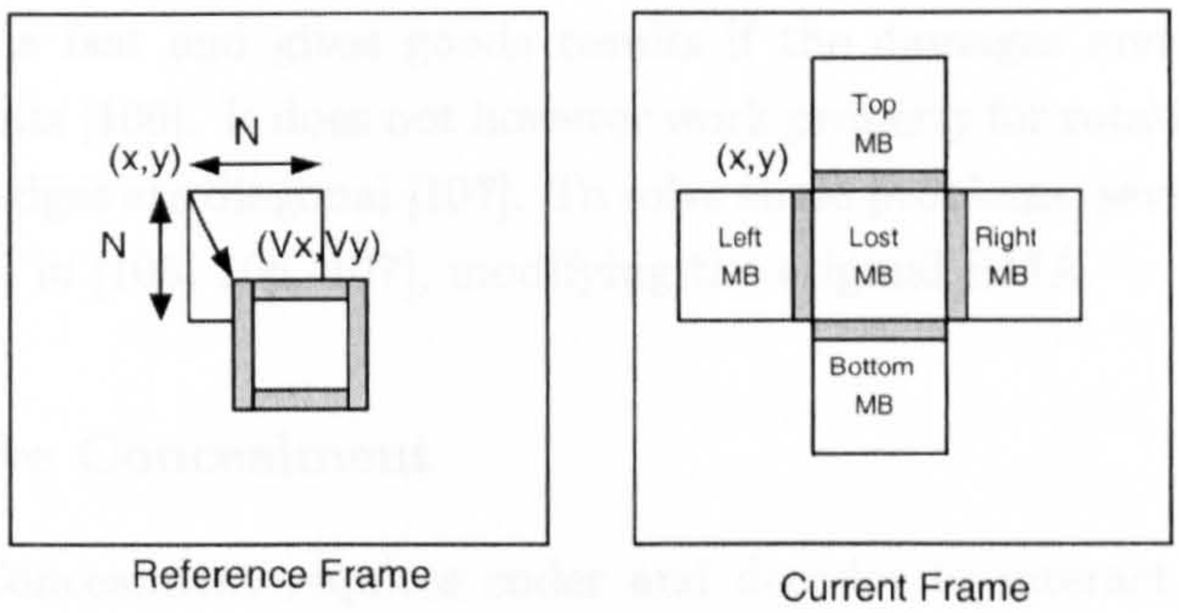


Figure 4.14: Boundary Matching Algorithm

4.3.3.4 Boundary Matching Algorithm (BMA)

This algorithm has been developed by Lam *et al.* in [104] and presents a solution to interpolate the MVs of the missing MBs [105]. It searches for the MV which satisfies best a spatial coherence criterion. The winning MV is the one minimising the boundary matching measure between the replaced MB and its surroundings [106].

BMA takes the lines of pixels, above, below, to the right and left of the lost MB in the current picture and uses them to surround each candidate from the previous decoded picture. The chosen MV (v_x, v_y) is found when the criterion given in equation 4.11 is minimised.

$$Criterion(v_x, v_y) = C_t(v_x, v_y) + C_l(v_x, v_y) + C_b(v_x, v_y) + C_r(v_x, v_y) \tag{4.11}$$

where C_t, C_r, C_b and C_r are the measures for the top, left, bottom and right boundaries respectively. They are given by:

$$C_t(v_x, v_y) = \sum_{i=0}^{N-1} \left[\hat{p}\,(x+i+v_x, y+v_y) - p\,(x+i, y-1)\right]^2 \tag{4.12}$$

$$C_l(v_x, v_y) = \sum_{j=0}^{N-1} \left[\hat{p}\,(x+v_x, y+j+v_y) - p\,(x-1, y+j)\right]^2 \tag{4.13}$$

$$C_b(v_x, v_y) = \sum_{i=0}^{N-1} \left[\hat{p}\,(x+i+v_x, y+v_y+N-1) - p\,(i, y+N)\right]^2 \tag{4.14}$$

$$C_r(v_x, v_y) = \sum_{j=0}^{N-1} \left[\hat{p}\,(x+v_x+N-1, y+j+v_y) - p\,(x+N, y+j)\right]^2 \tag{4.15}$$

where *(x,y)* is the coordinate of the top left pixel of the missing MB, $\hat{p}$ and p are the pixel values of the reference frame and the current frame respectively. N is the size of MB (or block) to conceal. Each of these four sums represents the boundary matching on a block side. For a simple implementation, the lost MV can be selected from predefined MVs such as zero MV, median and mean of MVs, [107].

This algorithm is fast and gives goods results if the damages areas involve only translation movements [106]. It does not however work properly for rotation or zooming [105] and when the edges are diagonal [107]. To solve these problems, several algorithms have been presented in [105, 106, 107], modifying the original BMA.

4.3.4 Interactive Concealment

Interactive Error Concealment requires coder and decoder to interact. Most of the time, interactive concealment techniques increase complexity but reduce the coding gain loss and achieve better performance [11, 102]. A feedback channel is required and the decoder can inform the coder about which part of the transmitted information is corrupted by errors. The coder may then adjust its transmission. One of the simplest techniques is ARQ (if supported) and consists on a simple retransmission of the corrupted data. This increases processing delays which may be unacceptable for real time applications [11]. The feedback channel can be used in the two following ways.

4.3.4.1 Reference Picture Selection (RPS)

When the encoder learns about the damaged parts of a previously coded frame, it can decide not to code the next P or B-frame with the most recent reference frame, but with an older reference picture which is known to be without error. Coder and decoder need therefore to store multiple frames. Information about the right reference picture to use is conveyed in the bit stream. This does not induce any extra delay at the decoder if the reference frame is not to far from the frame to be coded/decoded [11, 21].

4.3.4.2 Error Tracking based on Feedback Information

Instead of using a previous undamaged reference frame, the encoder can track how the damaged MBs in the frame would affect decoded blocks in the following and/or previous frames. MBs that will be affected by the original damaged parts may be processed in several different ways at the coder [11, 21]:

- They can be coded in intra mode to stop the propagation. This method is the simplest but suffers loss in coding gain (less efficient than the RPS).
- Error Concealment can be done at the encoder (similar technique as in the decoder since an encoder always contains a decoder) in order to match the reference picture for the next prediction coding. This technique is the most complicated.

This method requires lots of memory and is computationally intense if decoding is also performed at the coder side. There exists some low complexity algorithms that can be used to estimate the spatio-temporal error propagation in MB [21]. More about Error Tracking can be found in [11, 21].

4.4 H.264

The main goal of new H.264/MPEG - 4 part Advanced Video Coding (AVC) standard [36] jointly developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) is to enhance the coding efficiency of the codec as well as to provide a 'network-friendly' representation of the encoded video so that it is suitable for transmission over existing and future heterogeneous networks.

The development of the H.264/AVC standard (formerly called H.26L) follows the development of the previous H.261, H.263, H.263+ and MPEG - 4 standards. MPEG - 4 [108] has been standardised in 1998 and is aimed to provide a new level of interaction with visual contents. It provides technologies to view, access and manipulate objects rather than pixels with great error robustness for a large range of bit rates [109]. These technologies include shape and texture coding, motion estimation and compensation, as well as tools for error resilience.Full details on the coding techniques implemented in the H.263, H.263+ standards can be found in specific journal publications, conference proceedings and video coding books among which [90, 91, 92, 94, 110] are recommended. The key points are however recalled in the the next section.

A special issue of the IEEE Transaction on Circuits and Systems for Video Technology (July 2003, Volume 13) is dedicated to H.264 and gives an overview of the H.264. Specifically, [30, 34, 35] are recommended along with the book [111]. The specifications of H.264 can be found in [36].

4.4.1 Previous Standards: H.261, H.263 and H.263+

Following the development of H.261 [112] in the early 90s, ITU developed the H.263 [113] followed by H.263+ [114] in the early 2000s. These three standards use an hybrid bock diagram similar to MPEG-1/2 (see figure 4.3). H.261 differs in that the predicted frame at the output of the motion compensator is filtered with a *loop filter*. This filter modifies the prediction in order to attenuate the high frequencies. It has a blurring effect on the video and should only be activated for blocks with motion. Filtering is carried on a MB basis. The in-loop filter is optional in H.263 and H.263+.

The H.263 video codec standard has been designed for video coding for low and very low bit rate communication. The video is arranged in hierarchical structure with four layers: picture, group of blocks (GoB), MB and blocks, with a GoB being composed of $k \times 16$ lines pixels (k depends on the image format). The DCT-transformed and quantised components are represented in three dimensions by a (last, run, level) triplet and then VLC coded. The motion estimation in H.263 is carried on the luminance and at the MB level (as H.261 and MPEG-1) with a half-pixel precision. There is one MV per MB which is differentially coded with predictions taken from three surrounding MBs (left MB, top MB and top right MB) as shown in figure 4.15. Note some MVs might not be coded, depending on the motion. The decision whether or not a MB shall be motion compensated depends how on it can substantially reduce the prediction error.

The predictors are calculated separately for the horizontal and the vertical components of the MVs, MV1, MV2 and MV3. For each component, the predictor is the median value of these three MVs (specific rules apply for MBs at the border of a slice or a picture) and is given by:

$$pred_x = median\left\{MV_{1_x}, MV_{2_x}, MV_{3_x}\right\} \tag{4.16}$$

$$pred_y = median\left\{MV_{1_y}, MV_{2_y}, MV_{2_y}\right\} \tag{4.17}$$

The difference between the components of the current MV and the predictor are VLC coded. Their vector differences are defined by:

$$MVD_x = MV_x - pred_x \tag{4.18}$$

$$MVD_y = MV_x - pred_y \tag{4.19}$$

$$\tag{4.20}$$

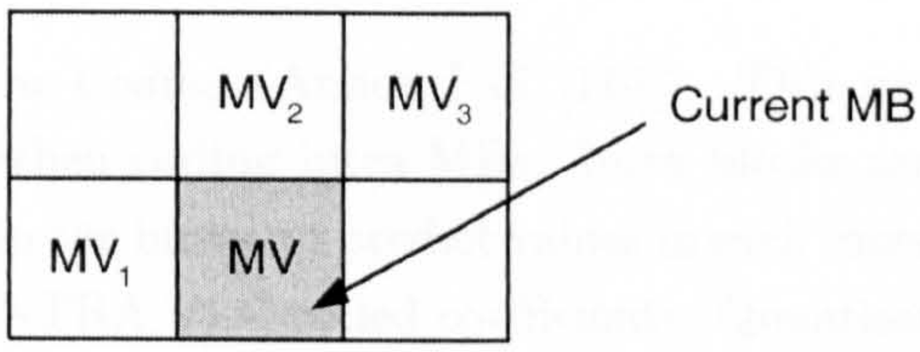


Figure 4.15: Motion Vector Prediction in H.263

Similarly to MPEG-2, H.263 adopts a layered bit stream syntax using Start-Codes. In addition to the basic video source coding algorithm, four negotiable coding options are included for improved performance [110, 113]:

- *Unrestricted Motion Vector*: MVs are allowed to point outside the picture.
- *Syntax-based Arithmetic Coding*: a binary arithmetic coding replaces the classing VLC coding.
- *Advanced Prediction*: this mode allows four MVs per MB (one per block) and *Overlapped Motion Compensation*, where each pixel of an 8×8 luminance prediction block is the weighted sum of three prediction values.
- *PB-frames*: A PB-frame consists of two pictures being coded as one unit. A PB-frame is composed of one P-picture (predicted from the previous P-picture or I-Picture) and one B-picture (predicted from both the previous P-picture and the current P-picture).

H.263+ development effort was intended for short-term standardisation of enhancements to the H.263 video algorithm for real-time applications [91]. H.263+ provides

enhanced video error resilience capabilities, especially for mobile video. The key features of H.263+ are the real-time audio-visual applications over error-prone channel [110]. The enhancements of H.263+ over H.263 fall into two categories [91]:

- Enhancement quality within existing applications: improving perceptual efficiency, reducing coding delay, providing greater resilience to bit error and data losses

- Broadening the current range of applications

H.263+ allows the use of a wide range of custom source formats, and hence opens a broader range of applications. Apart from the already existing features of H.263, H.263+ implements the following optional modes [110]:

- *Unrestricted Motion Vector Mode* (Annex D of [114]): This mode differs from the H.263 version and a reversible VLC (RVLC) is used to encode the difference motion vector, increasing resilience to channel errors. RVLC allows decoding the motion vector part of the bit stream in forward and reverse direction. Furthermore, the motion vector range is extended to up to [-256,+255].

- *Advanced Intra Coding* (Annex I of [114]): This mode improves compression performance when coding intra MBs. Intra blocks are predictively coded using nearby block in the image to predict values in each intra block. A separate VLC is used for the INTRA VLC coded coefficients. Quantisation of the DC coefficients for INTRA is also different. The prediction may be made from the block above or the block to the left of the current block being decoded. In addition to the zig-zag scanning, two other scans are available: alternate vertical and alternate horizontal. Alternate vertical is similar to the alternate scan in MPEG-2.

- *Deblocking Filter Mode* (Annex J of [114]): This mode introduces a deblocking filter inside the coding loop for the predicted frames similar to the one of H.261.

- *Slice Structured Mode* (Annex K of [114]): This mode implements a slice structure rather than a GoB structure. It allows the subdivision of the picture into segments containing variable numbers of MBs. It consists of a slice header followed by the coded MBs. The header acts as a resynchornisation point. The shape can vary (rectangular or not) and so can the order of transmission (sequential or arbitrary). Slice are independently encoded, meaning that no data dependencies across boundaries are allowed.

- *Supplemental Enhancement Information Mode* (Annex L of [114]): This mode allows supplemental information to be included in the bit stream in order to offer display capabilities.

- *Improved PB- frame Mode* (Annex M of [114]): this mode is an enhanced version of the PB-frame mode of H.263 and allows bi-directional, forward and backward prediction (H263 PB-frame mode uses only bi-directional prediction).

- *Reference Picture Selection Mode* (Annex N of [114]): This mode allows the selection of a reference frame other than the previous frame. This mode decreases the effect of error propagation and is used for error robustness.

- *Temporal , SNR and Spatial Scalability* (Annex O of [114]): As defined for MPEG-2 with minor differences. In H.263+, enhanced pictures can be predicted from the base layer picture and/or from previous enhanced picture.

- Other features include *Reference Picture Resampling Mode* (Annex P), *Reduced Resolution Update Mode* (Annex Q), *Independently Segmented Decoding Mode* (Annex R), *Alternative Inter VLC Mode* (Annex S), *Modified Quantisation Mode* (Annex T).

4.4.2 Video Coding Layer (VCL)

This layer defines the coding methods and many other new features used in the new H.264 standard. Each of these features taken separately does not significantly improve performance. However, when combined, the overall efficiency is greatly enhanced and they can achieve significant improvements. The range of applications is also wider when compared to prior standards: Broadcast over cable, satellite or DSL, Interactive or serial storage (DVD, optical or magnetic devices), conversational services over ISDN, Internet, Ethernet or LAN (wire or wireless), video on demand and multimedia streaming services, Multimedia Messaging Services [30, 34, 35]. As with all the previous standards, H.264 uses an hybrid video codec block diagram similar to the one of figure 4.3. The main difference is the insertion of a deblocking filter prior to the motion compensator.

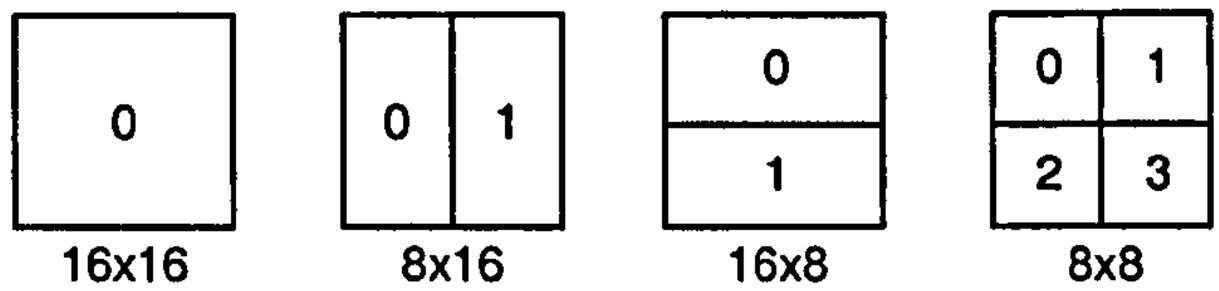
4.4.2.1 Improvements in efficiency

The various new coding tools that improve the efficiency can be separated into those improving the coding and prediction efficiency and those improving the robustness to errors.

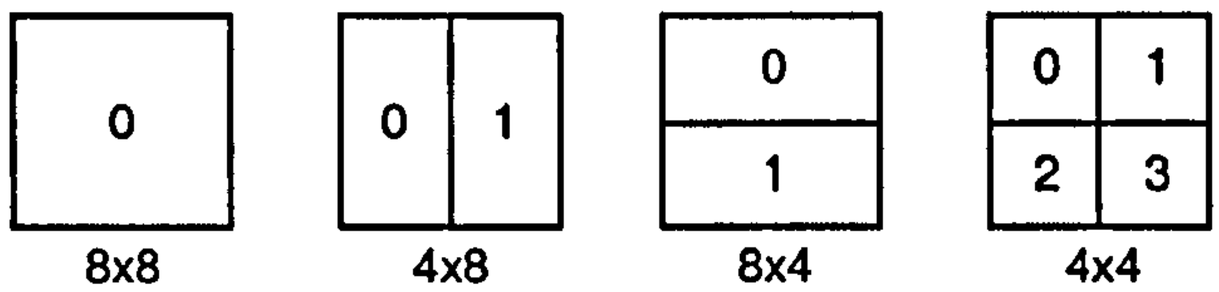
1. ***Prediction efficiency***

   The prediction efficiency has been improved with the following tools:

   - *Variable Block-size Motion Compensation*: more flexibility is added by allowing motion compensation blocks with various sizes. MBs are allowed to be segmented (or partitioned) into 16×8, 8×16 or 8×8 blocks and 8×8 blocks are allowed to be segmented into 8×4, 4×8 and 4×4 blocks for an improved motion compensation [35, 115, 116, 117] as shown in figure 4.16.
   - *Quarter-sample Accurate Motion compensation*: whereas most prior standards were using an half-pixel motion compensation accuracy, a quarter-pixel accuracy is implemented, improving the motion compensation. Note that the MPEG-4 standard uses a similar accuracy, but in H.264, the complexity is reduced [35].



(a) MB Modes



(b) 8x8 Modes

Figure 4.16: MB partitioning

- *Motion vector over picture boundaries*: the *Unrestriced Motion Vector Mode* option of H.263+, where motion vectors are allowed to point outside the picture area, is now included H.264 as a basic tool [35].
- *Multiple Reference Picture Motion Compensation*: the *Reference Picture Selection Mode* of H.263+ has proved to provide efficient coding. As basic tool, H.264 implements a multiple reference picture selection allowing the encoder to select the reference picture among a set of previously decoded pictures. This requires both encoder and decoder to store the reference pictures used for inter prediction in a multi-picture buffer [35, 116].
- *Weighted Prediction*: the motion compensation is allowed to be weighted.
- *In-loop Filtering*: this optional feature of H.263+ is now implemented as a basic tool in H.264. Applying an adaptive deblocking filter in the prediction loop reduces the block-based video coding artifacts and improved the resulting video quality.
- Others features improving the motion compensation include *Improved Skipped and direct motion inference, Direction spatial prediction for intra coding.*

2. ***Coding efficiency***

The coding efficiency has been improved with the following tools:

- *Small Block Size Transform*: H.264 adopts a 4×4 approximation of the DCT instead of the classic 8×8. This allows the decoder to represent signals in a more locally-adaptive fashion [35, 115, 116, 118]. Moreover, the scalar quantisation process uses 52 steps from 0 to 51 and an increase of 1 in the quantisation parameter decreases roughly the bit rate by 12%. Quantised coefficients are zig-zag scanned.
- *Hierarchical block transform*: while in most case the 4×4 DCT offers better benefits, for some pictures with blocks with large correlation a larger DCT

is preferable. H.264 allows therefore 8×8 and 16×16 DCT for intra coding [35]

- *Short word-length transform*: while other previous standards have generally required 32-bit processing for the complex processing of the transform computation, H.264 design requires only 16-bit arithmetic.
- *Exact match inverse transform*: while other previous standards use error tolerance bound for the transform causing drift between the encoder and the decoder and from one decoder to another, H.264 achieves an exact equality of decoded video between the encoder and the decoders.
- *Entropy coding*: H.264 supports two methods of entropy coding.
  - *Variable Length Coding* (VLC): For all the syntax elements except the quantised transformed coefficients, H.264 uses the default VLC entropy with a single infinite-extend codeword, thus only the mapping to the single codeword table is customised according to the data statistics. The single codeword table is an exp-Golomb code with very simple and regular decoding properties. For the quantised coefficients, a more elaborated code is adopted, using various VLC tables for the syntax elements depending on already-transmitted syntax elements. This is *Context-Adaptive Variable Length Coding*(CAVLC). With the tables well designed to match the corresponding conditioned statistics, the coding performances are greatly enhanced [35, 116].
  - *Context-Adaptive Binary Arithmetic Coding* (CABAC): The efficiency of the entropy coding is further improved if CABAC is used. The arithmetic code allows the assignment of non-integer number of bits to each symbol of an alphabet, especially for symbols with a probability greater than 50% [115]. The adaptive code permits adaptation to non-stationary symbols with changing statistics. Moreover, CABAC is context modelling, meaning that the statistics of already coded syntax elements are used to estimate future conditional probabilities [35, 116]. It is shown that CABAC increases the compression efficiency on average by 7-10% compared to CAVLC although CABAC is much more complex and can be very computationaly intensive [118].

Results in the literature [35, 116, 117, 118, 119] have shown up to 60% bit rate savings with the same visual fidelity at CIF format compared to various prior decoding schemes (MPEG-2, H.263+ and MPEG-4), due to the highly flexible motion model and the very efficient arithmetic-coding scheme. Figure 4.17 [116, 120] compares H.264 with MPEG-2, MPEG-4 and H.263 for the *tempete* sequence at CIF format.

4.4.2.2 Improvement of the robustness to errors

Because of the temporal and spatial predictions, an error in one frames propagates over the future frames. Common techniques exist to reduce the effect of the error

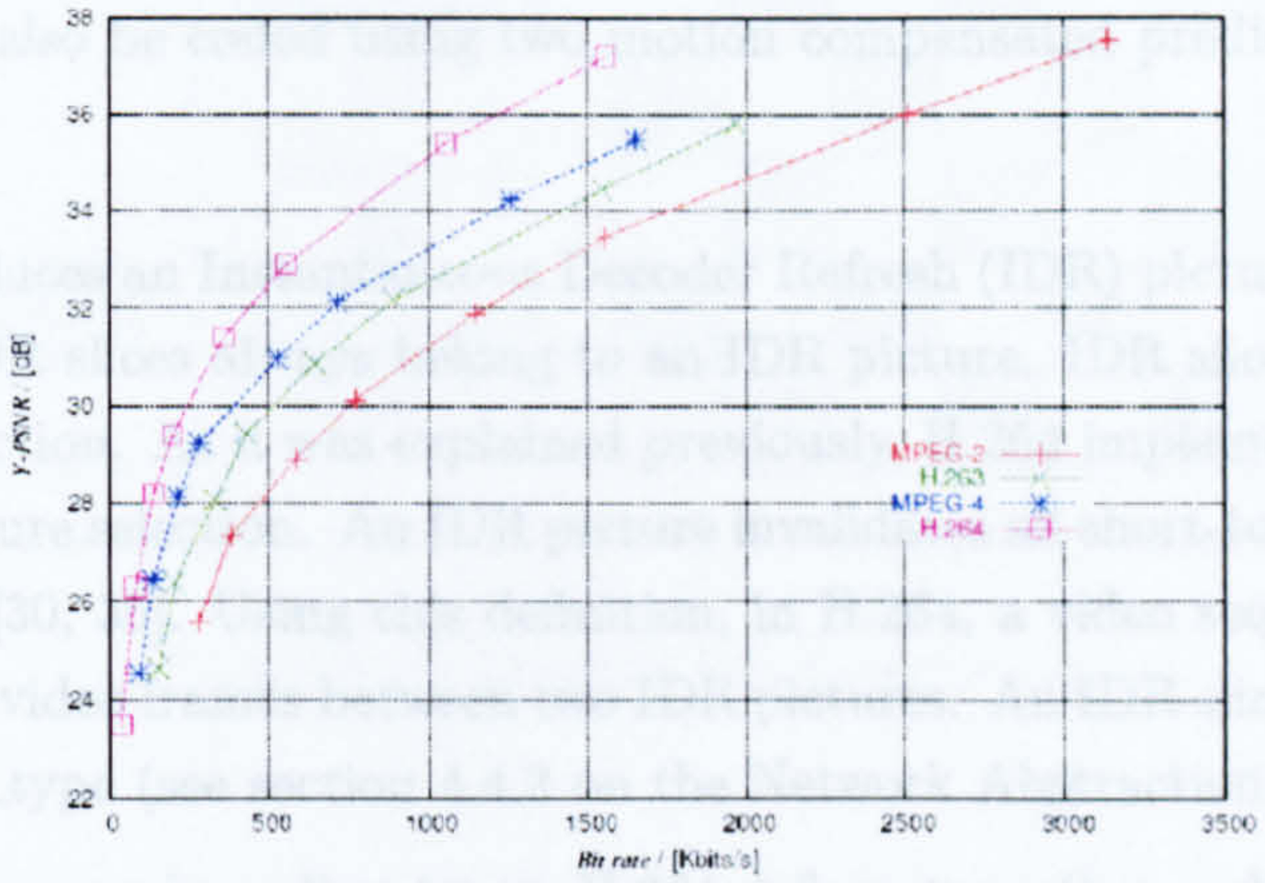


Figure 4.17: Rate Distortion Improvements

propagation either at the encoder or at the decoder, as shown in section 4.3. H.264 has been designed for storage but also for transmission over error-prone environment. Robustness to error is therefore a key feature of H.264 [30, 35, 121]. Here, the specific tools deployed by H.264 for error resilience and robustness are presented.

1. ***Multiple Reference Picture Structure***

   H.264 implements a multiple reference picture selection that improves the prediction efficiency. The reference picture is selected among a larger number of pictures that have been decoded and stored, improving the accuracy of the prediction. This features is similar to the error resilience tool explained in section 4.3.4.1 when encoder and decoder interact. In the case of a system with feed-back channel, the encoder can have knowledge whether one frame is corrupted or not and may use older but error-free reference MBs for prediction instead of using expensive intra coding [30].

2. ***Flexible Slice Structure***

   Similarly to MPEG-2 and H.263+, H.264 adopts a slice structure (a slice is a sequence of MBs which are processed in the order of raster scan within the slice). A picture may therefore be partitioned into one or several slices. Slices are self contained and are independent in the sense that they can be decoded without having knowledge of the other slices in the same pictures. H.264 has three main coding types for the slices:

   - *I slice*: where all the MBs of the slice are coded using intra prediction. A picture containing only I slice is a I picture.
   - *P slice*: where at least one MB is coded using motion compensation. Note that a P frame can contain a I slice and a P slice can contain MBs coded with intra prediction [35].

- *B slice*: In addition to the coding types available in a P slice, some MBs of a B slice can also be coded using two motion compensated prediction signals [35].

H.264 also introduces an Instantaneous Decoder Refresh (IDR) picture composed of IDR slices. IDR slices always belong to an IDR picture. IDR slices are coded using intra prediction. As it was explained previously, H.264 implements a multiple reference picture selection. An IDR picture invalidates all short-term reference memory buffers [30, 35]. Using this definition, in H.264, a video sequence corresponds to all the video frames between two IDR pictures. An IDR slice is detected with its nal_unit_type (see section 4.4.3 on the Network Abstraction Layer)

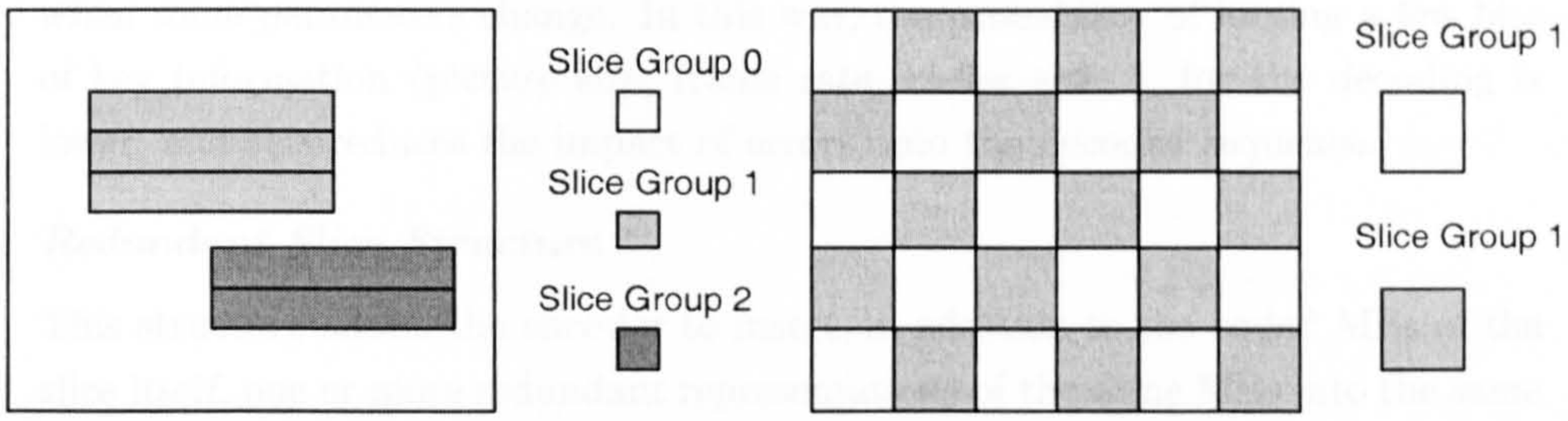
In addition to these main coding types, H.264 defines two other coding types for SI and SP slices, explained in point number 6 in section 4.4.2.2. The type of the slice is given in the slice header and its value is given in table 4.1 [36]. The slice type values between 5 and 9 specify that all the other slices of the current picture shall have the same type. For an IDR slice, the slice type should be either, 2, 4, 7 or 9. The slice type of a non IDR slice is either 0, 1, 2, 5, 6 or 7.

Table 4.1: Slice Type values

slice_type	*Slice*	*slice_type*	*Slice*
0	P slice	5	P slice
1	B slice	6	B slice
2	I slice	7	I slice
3	SP slice	8	SP slice
4	SI slice	9	SI slice

H.264 introduces a new approach with the *Flexible MB Ordering* (FMO) using the concept of a *Slice Group*. Each slice group is a set of MBs defined by a *MB to slice group map*. The map consists of a slice group identification number for each MB in the picture specifying which slice group the it belongs to [35]. One slice group can be partitioned into one or more slices, such that a slice is a sequence of MBs within the same slice group. Note that MBs are still processed in raster scan within the slice and MBs within one slice are not necessarily adjacent anymore. Figure 4.18(a) shows an example of FMO with three slice groups. Slice group 0 contains one slice (background), slice group 1 contains three slices and slice group 2 is composed of two slices. Figure 4.18(a) shows an example of FMO with two slice groups known as the 'chess-board' or'dispersed' type.

With the use of FMO, the visual impact of slice-loss is reduced, especially if the chess-board pattern is used. Since every lost MB has several spatial neighbours belonging to the other slice group, an error concealment mechanism has a lot of information it can employ to recover the missing MB. It also allows unequal error protection with slice groups containing important information (e.g. foreground)



(a) Example 1 (b) Example 2: Chess-board type

Figure 4.18: Flexible MB Ordering

being allocated more protection. The price paid for the use of FMO is a lower coding efficiency [30]. In order to preserve the independence of each slice, the motion compensation efficiency has to be reduced because MBs within one slice might not be adjacent anymore. Note that a slice is constrained by its length either in terms of bytes or in terms of MBs. One slice can eventually be the full picture (or slice group).

3. ***Intra Placement***

   To combat drifting effects, the intra elements can be inserted. These elements can either be MBs, or slices or pictures and are intra coded, without reference to any encoded pictures. They provide re-synchronisation and can stop error propagation. H.264 has two forms of slices that contains intra MBs: Intra slice (I-slice) and Instantaneous Decoder Refresh (IDR) slices. Because it invalidates all short-term reference memory buffers, an IDR picture has a stronger re-synchronisation property than a frame containing only I slice. An I frame would cancel the drift for the duration of the picture only but as multiple reference frames are used, future frames may reference frames prior to the I frames and may re-establish the drift, even in an error-free environment [30]. Random or pseudo-random intra MB can also be inserted [122] and yields to a relatively small bit rate penalty [121].

4. ***Parameters Sets Structure***

   A parameter set contains information that is expected to rarely change [35] and that can be applied to a large number of pictures [111]. There are two kinds of parameter sets. Sequence parameter sets contain all the information related to a sequence of pictures, and picture parameter sets contain all the information related to all slices belonging to a single picture. Multiple different sequence and picture parameters sets can be available at the decoder. The encoder chooses the appropriate picture parameters to use by referencing it in the slice header. The picture parameter set itself contains a reference to the sequence parameters set to be used [30]. These sets need to be transmitted first either out-of band or in-bound with an appropriate protection and are only required to be retransmitted

when some parameters change. In this way, the probability of loosing a few bits of key information (picture size, frame rate, buffer size...) for the decoding is lower, and this reduces the impact of errors onto the decoded sequence.

5. ***Redundant Slice Structure***

   This structure allows the encoder to insert, in addition to the coded MBs of the slice itself, one or more redundant representations of the same MBs into the same bit stream. The key difference with other redundancy schemes is that the redundant slices can be coded with different coding parameters such as different QPs. The original slice is known as the primary slice and can have a good visual resolution (low QP) whereas the redundant slice can have a less accurate resolution (high QP) with fewer bits. The decoder typically reconstructs the primary slice, however, if this primary slice is lost, the redundant slice is reconstructed [30].

6. ***Data Partitioning***

   Usually, all symbols of a MB are coded together in a single bit stream to form a slice. Already implemented in MPEG-2, data partitioning creates more than one bit stream (called a partition) per slice [30, 35]. H.264 implements a three level data partitioning structure. The first level (Type A partition) contains all the header information, including MB types, QPs and MVs [111]. This first level is therefore the most important. Without it, the two other levels can not be used. The second level (Type B partition) carries Intra coding block pattern (CBP) and intra coefficients [111]. It requires the availability of the Type A partition to be useful. The third level (Type C partition) contains the inter information (Inter CBP and Inter coefficients). This partition is the least important since the data it carries do not contain re-synchronisation information. To be useful, Type C partition requires the availability of Type A and Type B partitions [30]. Note Data partitioning is not supported for IDR pictures.

   At the receiver, all the partitions are required for decoding. However, if a Type C partition is missing, Type B and Type A are still useful and can improve the efficiency of the error concealment. In a similar way, if Type B and Type C partitions are missing, Type A is still useful for the concealment since it can provide the MB types and the MVs [30].

7. ***Other Features***

Among other features used for error resilience, the following are highlighted:

- *Arbitrary Slice Ordering* (ASO): H.264 has the capability of sending and receiving slices in any order relative to each other. This can improve the end-to-end delay for real-time applications [30].
- *SP/SI synchronisation/switching Pictures*: The main features of this structure is that identical SP-frames can be reconstructed even when different reference frames are used for prediction. SP-frames make use of motion

compensated predictive coding to exploit temporal redundancy in the sequence while still allowing identical reconstruction of the frame, even when different reference frames are being used [30, 123]. They are used for efficient switching between bit streams coded at various bit-rates [116, 117].

- *Spare Pictures*: This structure allows to indicate similarities between a reference picture and other pictures. It helps to improve the quality of some decoded pictures by replacing the references pictures by better ones, to improve error concealment by concealing a lost block as a corresponding block in one of the spare picture [124].
- *Scalability*: The H.264 task group has released a call of proposals on scalable video coding. The propositions presented at the meeting in Munich in March 2004 are still under discussion.

The evaluation of these various error resilience features can be found in [34, 121, 122, 125, 126, 127].

4.4.2.3 Profiles and Levels

Similarly to H.263+ and MPEG-2, H.264 defines *Profiles* and *Levels* in order to supply conformance points for inter-operability between various applications [35, 116]. All the coding tools are not required for all the applications [120]. For example, the error resilience tools are not required for error free networks or/and transmission [117]. Therefore subsets (Profiles) of coding tools with different classes of applications are defined. Three profiles have been developed: Baseline (with basic features), Extended (the most complete) and Main (without error resilience). All decoders conforming to a specific profile must support all features in that profiles. Encoders are not required to make use of any particular set of features supported in a profile but have to provide conforming bitstream [35].

Because of the various processing powers and memory sizes of decoders (real-time or non real-time), levels provide constraints mainly on picture sizes (from QCIF to 4000x2000), frame rates (from 15 to 120 frames per second), bit rates (from 64kbits/s to 240Mbits/s) and number of reference pictures (from 3 to 9) [35, 117, 120].

4.4.3 Network Abstraction Layer (NAL)

The *Network Abstraction Layer* (NAL) is the interface between the VCL layer and the underlying layers [128] as shown in figure 4.19. It provides a means to transport the video over numerous and heterogeneous networks by allowing a seamless and easy integration of the coded video stream into current and future system architectures [128]. It operates on a NAL unit (NALU) basis.

4.4.3.1 NAL Unit (NALU)

A NALU is a byte string consisting of a 1-byte header followed by a payload of variable length and is the basic output of the encoder. A NAL unit is depicted in figure 4.20. If

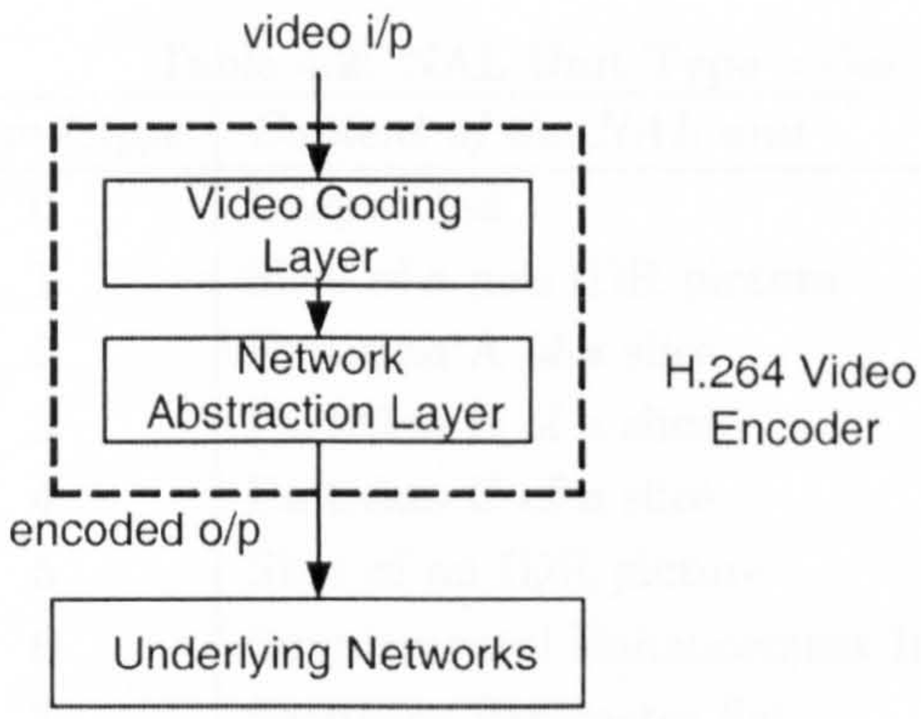


Figure 4.19: H.264 Encoder

it does not carry side information (parameter sets or SEI), a NAL unit always carries one slice (or one partition of a slice). The header is composed of:

- *forbidden_bit* (1 bit): set to zero and can be flagged by underlying media-aware networks to indicate to the decoder that the content of the NAL unit is erroneous. The decoder can either drop or conceal the corrupted slices [35, 129].
- *nal_ref_idc* (2 bits): specifies whether the content of the NAL payload is allowed to be part of a reference picture or not.
- *nal_unit_type* (5 bits): specifies the type of data structure contained in the payload [36]. The different nal unit payload types are presented in table 4.2.

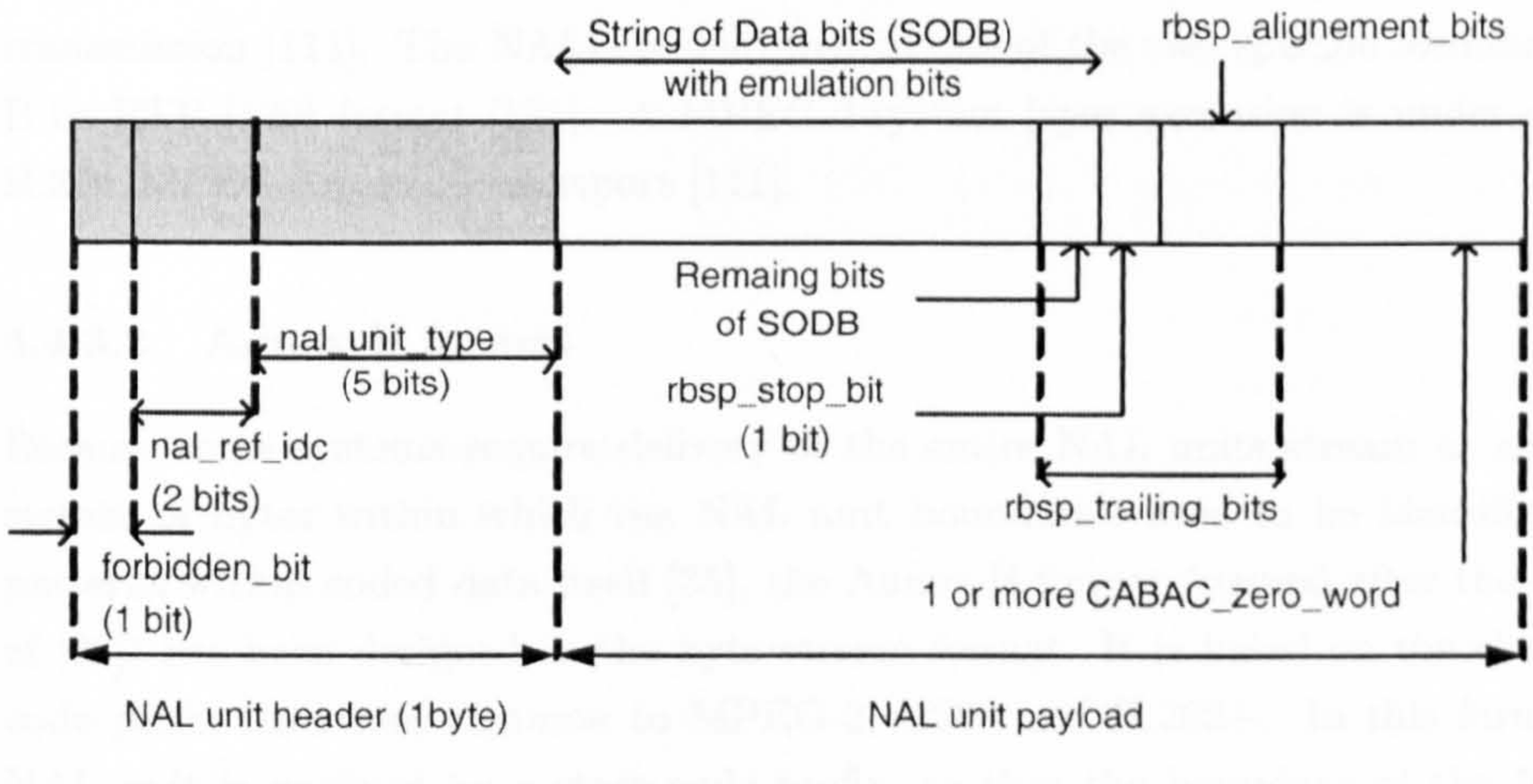


Figure 4.20: NAL Unit

The String Of Raw Data Bits (SODB) coming from the AVC encoder is first finalised by adding the Raw Bits Sequence Payload (RBSP) trailing bits (*rbsp_trailing_bits* composed of the remaing bits of SODB, a stop bits equal to 1 and the bytes alignment bits). One or more zero bytes are added at the tail when CABAC is used to form the

Table 4.2: NAL Unit Type codes

nal_unit_type	*Content of the NAL unit*
0	Unspecified
1	Slice of a non IDR picture
2	Partition A of a slice
3	Partition B of a slice
4	Partition C of a slice
5	Slice of an IDR picture
6	Supplemental Enhancement Information
7	Sequence Parameter Set
8	Picture Parameter Set
9	Access Unit Delimiter
10	End Of Sequence
11	End of Stream
12	Filler Data
13-23	Reserved
24-31	Unspecified

RBSP bits. RBSP bits are then interleaved as necessary with *emulation prevention bytes* inserted to prevent a *start code prefix* from being accidentally generated inside the payload. When the 0x00 0x00 000000XX pattern is found (XX is either 00, 01, 10 or 11), the emulation prevention byte 0x03 is inserted to form: 0x00 0x00 00000011 000000XX [36].

Note that there is no mandatory transport mechanism for H.264 coded data. However, there are a number of possible transport solutions depending on the methods of transmission [111]. The NALU is delivered in one of the two specific formats: Annex B or RTP [130] format [131]. A MPEG-2-system layer extension is under study for H.264/MPEG-4 part 10 transport [111].

4.4.3.2 Annex B format

Because some systems require delivery of the entire NAL units stream as an ordered stream of bytes within which the NAL unit boundaries need to be identifiable from patterns within coded data itself [35], the Annex B format (named after the Annex B of [36]) has been designed as the byte stream format. It is based on the classic start code prefix insertion, common to MPEG-2, H263 and H.263+. In this format, each NAL unit is prefixed by a start code prefix, so that the boundary of the NAL unit can be found by the decoder by searching for it in received bytes stream. The use of emulation bytes guarantees that start code prefixes are the only identifiers of the NAL unit boundaries [35].

Depending on the importance of the NAL unit, the start code prefix differs. If the NAL unit carries either parameter sets (picture or sequence) or the first slice of a picture, the start code prefix is then four byte long and is equal to 0x00000001.

Otherwise, the start code prefix is only three byte long and is equal to 0x000001 [36].

4.4.3.3 Packet-based format

Other system, such Internet or IEEE 802.11 require data to be transmitted in packets that are framed by the transport protocol. NAL unit boundaries are then easily identifiable by the decoder for re-synchronisation without the use of start code prefixes. Nevertheless, even if they are not required in the packet-based format, the emulation bytes are however inserted. Since the emulation prevention is done prior the choice of the transport format, emulation bytes are present in both case. The packet-based format developed for the transport of H.264 is the Real-time Transport Protocol (RTP) [130, 131] with dedicated rules. This RTP format is fully detailed in section 6.3.1 for the case of the study of the packetisation.

4.4.4 Advanced Error Concealment (AEC) Algorithm

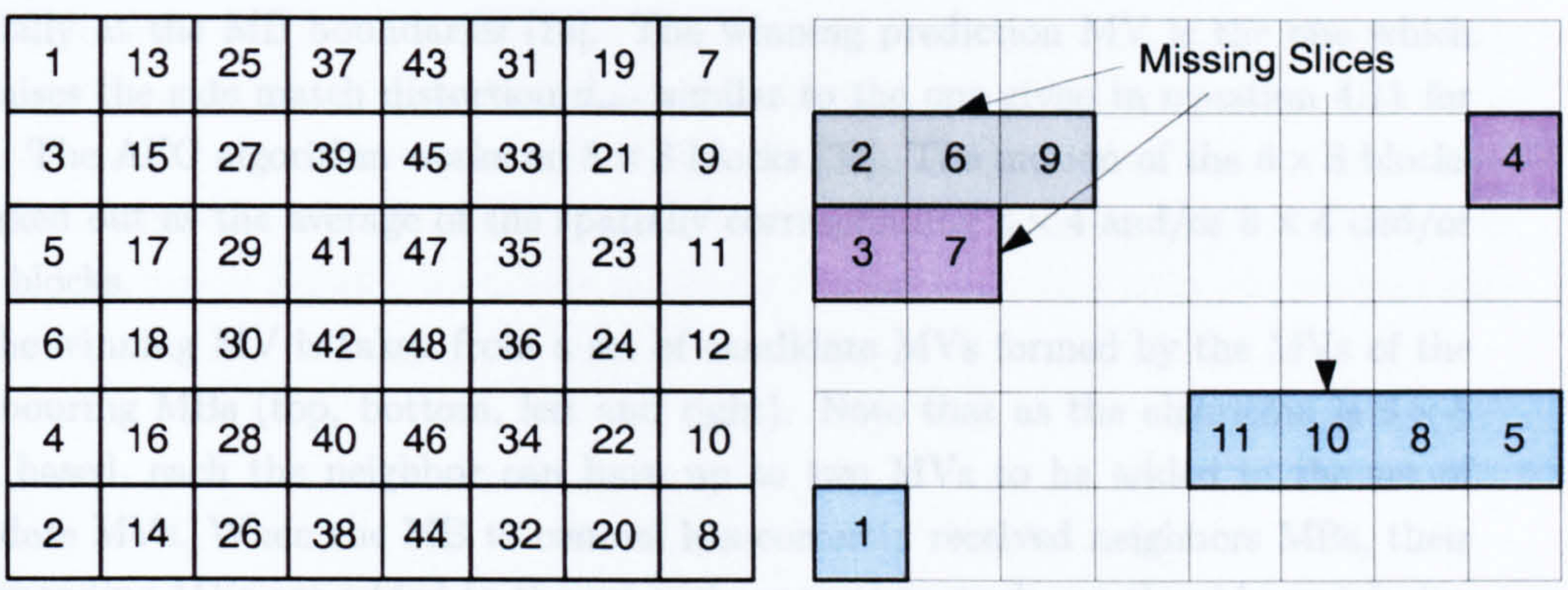
The design of a specific error concealment algorithm is outside the scope of H.264. The authors of the specification and of the test model software [132] however recommend a non-normative algorithm [16, 133] in order to provide a basic level of error resilience for the decoder. Any error-robust improved coding or error concealment algorithm shall therefore be compared against the H.264 test model equipped with this algorithm [34]. This algorithm is called Advanced Error Concealment (AEC) and is included as an informative part of the test model description.

It is assumed that erroneous slices are not fed to the decoder but discarded before decoding. Neither error checking nor bit-integrity detection is performed at the decoder. The correctly received slices are first decoded. The missing slices of one frame are then concealed using a *MB map*, stating if one MB has been correctly received or lost [16].

The order of processing differs from the classic left to right and top to bottom order. The frame is scanned column-wise from the left and right edges to the center of the image as shown in figure 4.21(a) on the example of frames with 8 by 6 MBs. Consecutive lost MBs in a column are concealed starting from the top and bottom of the lost area toward the center of the areas as shown in figure 4.21(b) where three slices are missing. This processing order allows better concealment of the usually hard to conceal central part of the frame where most of the motion takes place. This ensures that lost MBs at the center of an image are concealed using as many neighboring concealed MBs as possible [34].

The INTRA concealment is based on weighted pixel value averaging. The weight used for averaging is simply the inverse distance between the source and destination pixels as shown in figure 4.4.4. Only MBs correctly received are used for this concealment if at least two of them are available. Otherwise already concealed MBs are used.

The INTER concealment uses a boundary-matching-based motion vector recovery similar to the BMA algorithm detailed in section 4.3.3.4. Depending on the motion activity (average MVs) of the correctly received slices in the frame, different conceal-



(a) Scanning order

(b) Concealment order

Figure 4.21: Order of Concealment

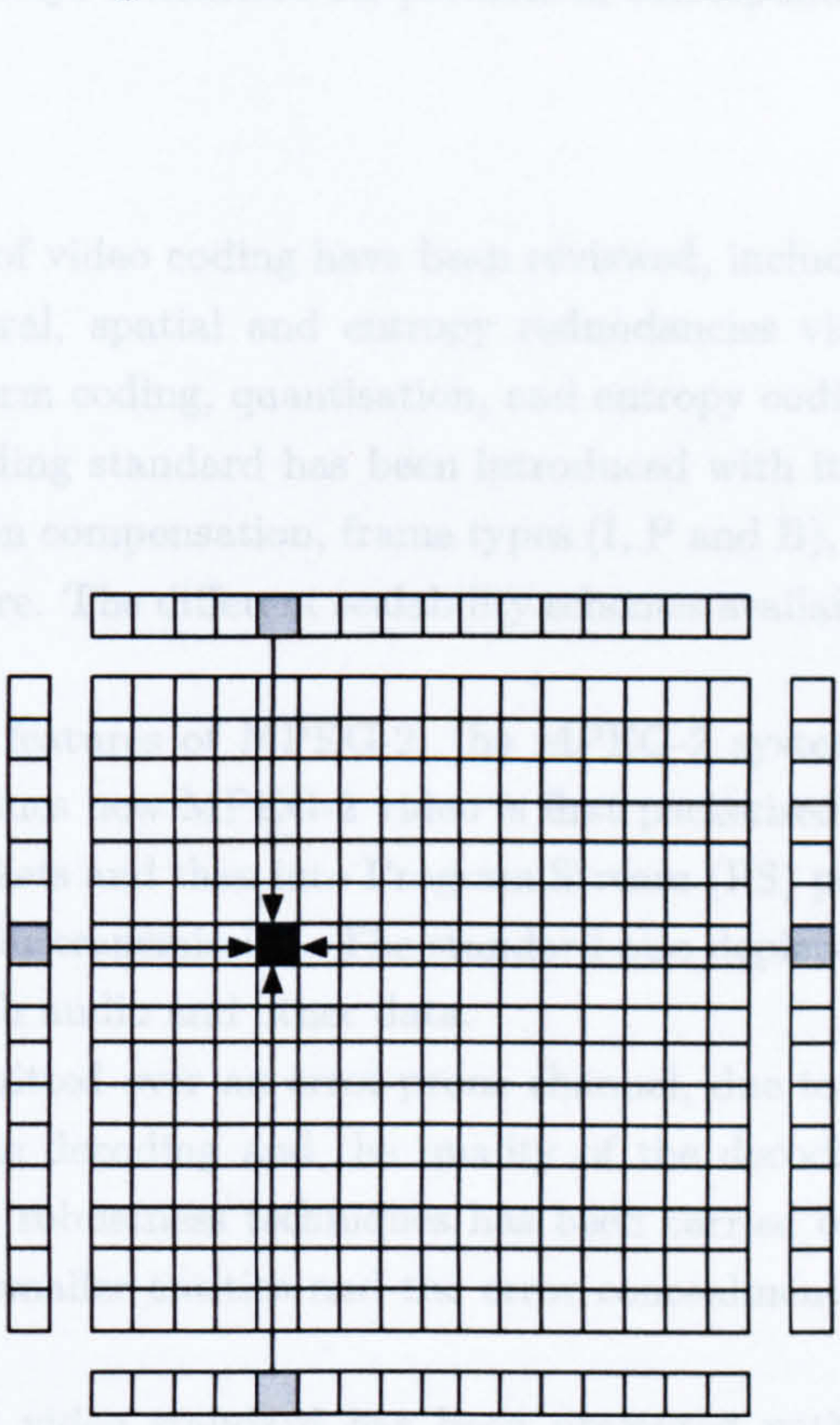


Figure 4.22: Intra Spatial Concealment based on weighted pixel average

ments take place. If the activity is smaller than a threshold, a simple copy of the co-located MBs in the reference frame is performed on the missing slices. Otherwise, motion compensated error concealment is used [16, 133]. The MV of the missing MBs is predicted from spatial neighbours. The decision on which neighbor motion vector to use as a prediction, is made based on the smoothness of the reconstructed image, especially at the MB boundaries [16]. The winning prediction MV is the one which minimises the side match distortion d_{aec} similar to the one given in equation 4.11 for BMA. The AEC algorithm works on 8×8 blocks [34]. The motion of the 8×8 blocks is worked out as the average of the spatially corresponding 4×4 and/or 8×4 and/or 4×8 blocks.

The winning MV is taken from a set of candidate MVs formed by the MVs of the neighbouring MBs (top, bottom, left and right). Note that as the algorithm is 8×8 block based, each the neighbor can have up to two MVs to be added in the set of candidate MVs. When the MB to conceal has correctly received neighbors MBs, their corresponding MVs are added to the set and are used to work out the side match distortion with their corresponding reference pictures . Otherwise, concealed neighbours MBs are considered [16]. Note that the zero motion vector is always present in the set of candidate MVs and is always considered for prediction, corresponding then to PFC.

4.5 Conclusions

In this chapter, the basics of video coding have been reviewed, including a description of the reduction of temporal, spatial and entropy redundancies via motion prediction/compensation, transform coding, quantisation, and entropy coding respectively.

The MPEG-2 video coding standard has been introduced with its coding features such as bi-directional motion compensation, frame types (I, P and B), 2D VLC entropy coding, and layered structure. The different scalability schemes available with MPEG-2 have also been presented.

Along with the coding features of MPEG-2, the MPEG-2 system layer has been described. This system defines how MPEG-2 video is first packetised into Packet Elementary Stream (PES) packets and then into Program Stream (PS) packets for storage or Transport Stream (TS) for transmission. The standard also depicts how video packets may be multiplexed with audio and other data.

When a video is transmitted over an error-prone channel, due to the compression techniques, errors affect the decoding and the quality of the decoding video. A review of error resilience and robustness techniques has been carried out, including the partitioning of frame into smaller entities and the error concealment Block Matching Algorithm.

Finally, the new H.264 video standard has been presented with its new coding features. H.264 has been designed so that the compression efficiency (coding and prediction) at Video Coding Layer (VCL) is greatly improved with the combinations of various techniques. H.264 has also been designed so that it can be easily transmitted over many various systems. This section has included details on the error resilience

and robustness features, among them the Flexible MB Ordering (FMO) and multiple reference pictures selection can be highlighted. The VCL is then linked with a network-friendly layer, the Network Abstraction Layer (NAL) providing an interface with underlying networks. The concealment algorithm used as reference by the authors of the specifications and of the test model software is also presented.

Due to its compression efficiency that out ranges the previous MPEG-2 and H.263 standards, and due to its error robustness features, H.264 seems adapted to provide reliable transmission. Moreover its flexibility with the Network Abstraction Layer allows transmission over various packet-based systems or bytes stream orientated networks. This thus makes H.264 a candidate for future transmission over networks such as wireless LANs.

Chapter 5

Video Transmission Enhancement over the IEEE802.11a PHY layer and the IEEE 802.11 MAC

This chapter investigates the use of Forward Error Correction (FEC) and Space-Time Block Codes (STBC) on the PHY layer to minimise the use of the MAC layer ARQ for video transmission. It is organised as follows: section 5.1 introduces the chapter and defines the requirements and challenges of MPEG-2 transport stream video transmission over the IEEE 802.11a/g PHY and the IEEE 802.11 MAC layer without strong reliance on ARQ as explained in chapter 3. Section 5.2 studies the use of Reed-Solomon codes implemented in a concatenated code to reduce the PER. Section 5.3 describes STBC spatial diversity techniques to enhance the PHY layer. It also presents performance results for such techniques. Section 5.4 studies a system combining RS concatenated coding with an enhanced Space Time Block Code PHY layer. Performance results in terms of PER, BER, use of ARQ, delay and throughput are presented in this section. Finally, section 5.5 concludes the chapter.

5.1 Introduction

Recently, there has been a growing interest in video streaming over Coded Orthogonal Frequency Division Multiplexing (COFDM) based Wireless Local Area Networks (WLANs), such as IEEE802.11a/g and Hiperlan/2 [134, 135, 136]. They provide data rates at layer 1 up to 54Mbits/s in a 20MHz bandwidth and are a possible solution for home entertainment, video on demand and other home multimedia communication products. Real-time video transmission is bounded by delay constraints and requires very low end-to-end delay [137]. As detailed in chapter 3, Retransmission with Automatic Repeat reQuest (ARQ) introduces delay and its use should be avoided or at least limited. ARQ is however a mandatory feature of the IEEE 802.11 MAC layer as explained in chapter 3 and is used to correct errors [80, 138]. In chapter 3, the effect of the number of ARQs on delay was studied and packets were seen to suffer 20 ms

of delay when a maximum of 8 ARQs per packet was allowed. For delay purposes, and in order to guarantee a video transmission without strong reliance on ARQ, the IEEE 802.11a/g performance may need to be enhanced so that the PER is reduced and the number of MAC retransmissions is limited. In this chapter, it is assumed that retransmission only happens when a packet is corrupted and that no collision occurs, corresponding to 'lightly loaded' network scenario.

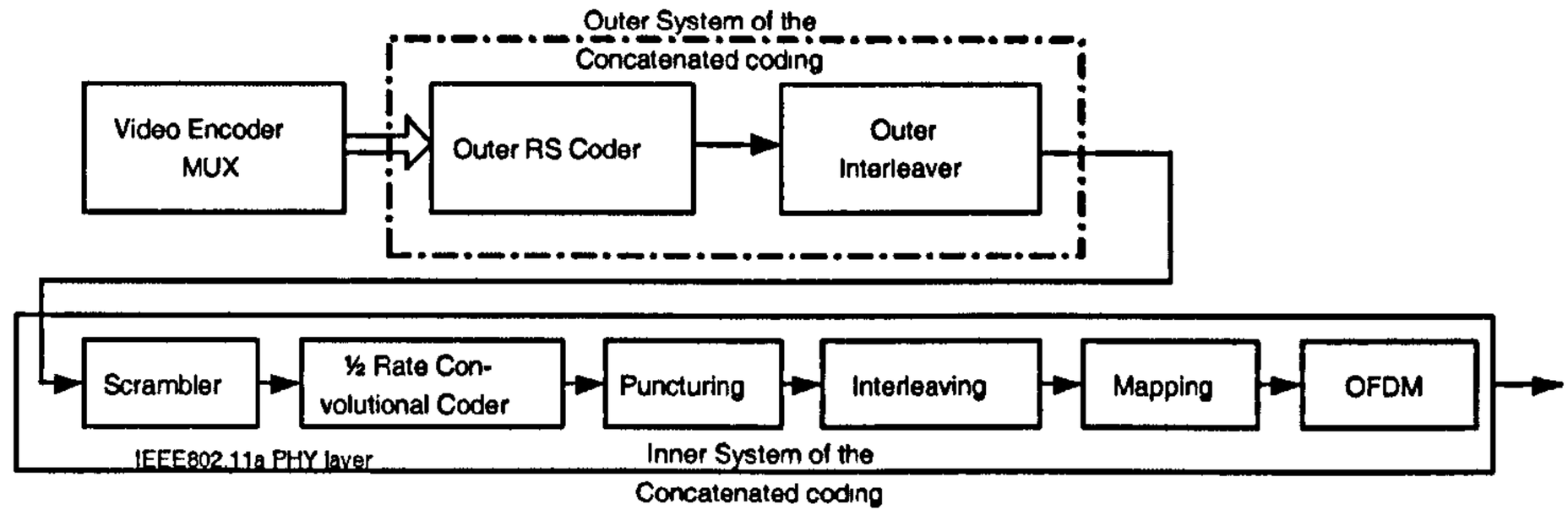
An alternative to ARQ and other error-resilience techniques is the use of Forward Error Correcting (FEC) codes [137]. FEC codes are commonly employed to add robustness and to improve the PER/BER performance. More specifically, Reed Solomon codes (RS) [139] are particularly adapted for correcting burst of loss packets [140]. RS codes concatenated with convolutional codes have already been implemented in Digital Video Broadcasting Terrestrial and Satellite (DVB-T and DVB-S) [32, 141] systems, where ARQ is not employed. DVB uses the MPEG-2 system [99] and allows several data streams to be multiplexed [138]. MPEG-2 Transport Stream [100] was described in section 4.2.2. DVB-T and S use MPEG-2 TS packets of 188 bytes and are deployed in outdoor environments that generally have Line of Sights (LOS). The study conducted in this chapter is following the DVB system but uses WLANs at 2.4 and 5.2GHz deployed in home and office environments with Non-LOS (NLOS).

In order to enhance further the radio performance of the PHY layer, multiple transmit and receive antennas can be used to provide diversity. A simple and very attractive form of Space Time Block Coding (STBC) was proposed recently by Alamouti in [142] and generalised by Tarokh in [143]. It only requires a small degree of additional complexity and is suitable for the low mobility environments in which WLANs will be deployed. STBC can enhance performance by exploiting spatial diversity, rather than by using temporal diversity (retransmission). This chapter investigates the combination of both systems (concatenated coding and STBC) in order to enhance the performance, i.e. the PER of the PHY layer so that the transmission of MPEG-2 streams do not strongly rely on ARQ [138].

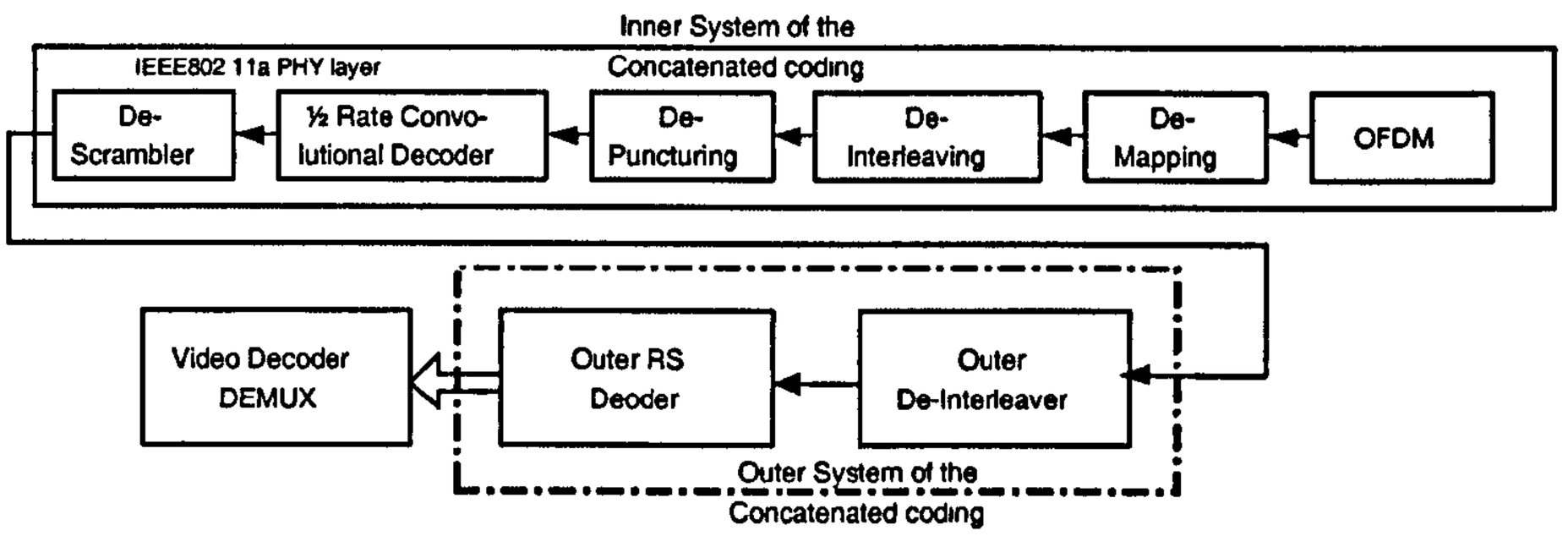
5.2 Use of Concatenated Coding Reed Solomon (outer code) with IEEE802.11a PHY

To reduce the use of ARQ at the MAC level, the PER at the physical layer should be improved, so that fewer packets require retransmission. This can be achieved by implementing FEC at the physical layer. The IEEE 802.11a/g PHY already employs channel coding, however it can be combined with another outer code to perform a concatenated code with the inner code provided by IEEE 802.11a/g PHY layer. Concatenated codes are commonly used in conjunction with outer and inner interleavers. This section examines the use of concatenated coding as implemented in DVB-T and DVB-S [32, 141] for MPEG-2 TS packets of 188 bytes. Both schemes implement a transport multiplex adaptation and randomisation for energy dispersal, an outer code using a RS code, a convolutional interleaver, and an inner punctured code from the

WLAN PHY layer [32, 141]. RS are popular for transmission over bursty channel since it provides strong correction capability particularly adapted for correcting burst of loss packets [140]. Figure 5.1 shows the functional block diagram used for the study.



(a) Transmitter



(b) Receiver

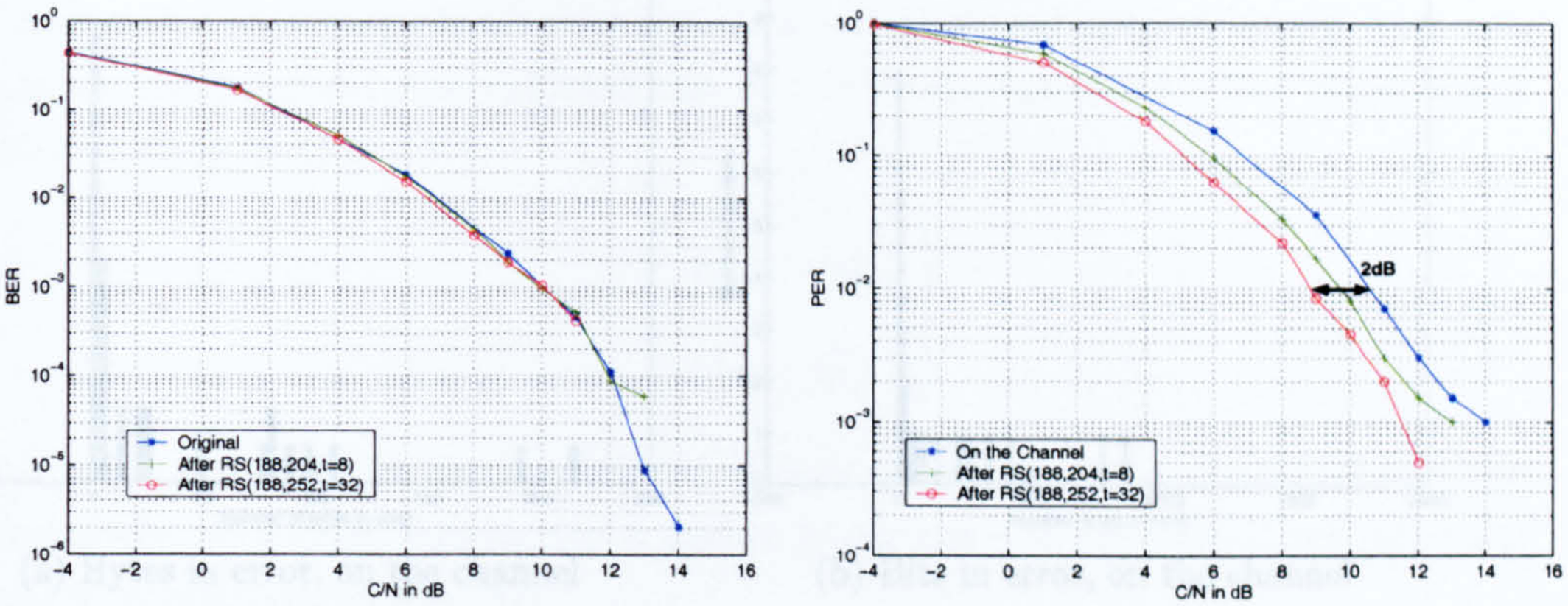
Figure 5.1: Block Diagram of a Concatenated Code for MPEG-2 TS packets transmission

DVB-S and DVB-T employ a shortened Reed Solomon (RS) encoder for the outer code (RS (188, 204, t=8) correcting up to 8 bytes) and a convolutional interleaver. A similar structure is adopted here, employing a RS code with two different correction capabilities of 8 and 32 bytes for study. Table 5.1 shows the different parameters used for the simulations and the corresponding coding rate and the video bit rate reduction due to the coding overheads. Decoding has been performed with the classical Berlekamp-Massey algorithm using hard-decisions [139].

Table 5.1: Reed Solomon Coding Parameters

Code	*Input length in bytes*	*Output length in bytes*	*Correction Capability*	*Bit Rate Reduction*	*Coding Rate*
No RS	188	188	0 bytes	0%	1
RS	188	204	8 bytes	7.84%	0.921
RS	188	252	32 bytes	25.39%	0.746

The BER and PER performance for mode 1 over the IEEE 802.11a/g standards is shown in figure 5.2 for RS(188,204,t=8) and RS(188,252,t=32). The channel used for these simulations is the ETSI - Channel A (NLOS offices with an rms delay spread

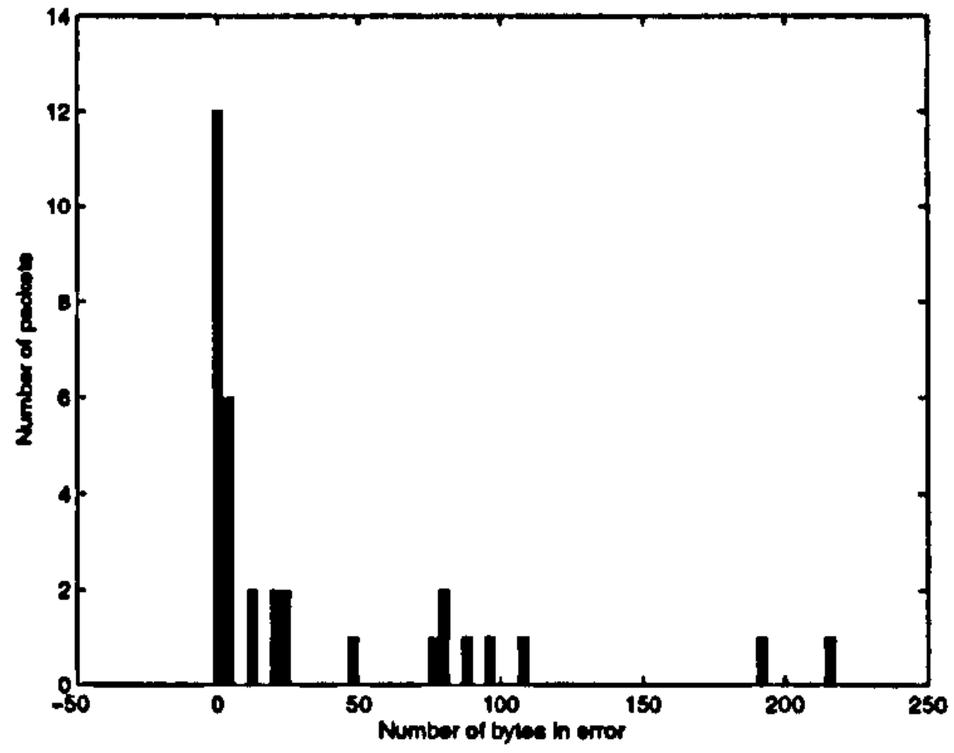


(a) BER (b) PER

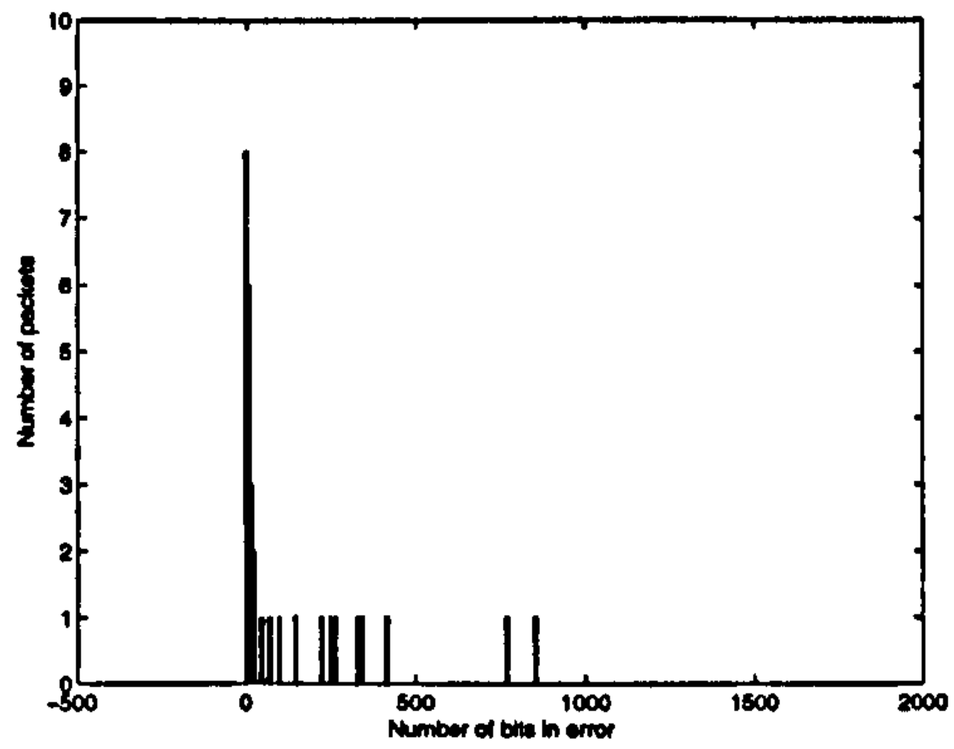
Figure 5.2: Performances for mode 1 with RS(188,204,t=8) RS(188,252,t=32) over IEEE802.11a

of 50ns). We observe that, although the PER performance is improved (2dB gain for the 32 byte correction capability case at a PER of 10^{-2}), the BER performance is not enhanced. The BER after the outer coder follows closely the BER after the inner coder. This is due to the fact that the outer convolutional decoder can correct packets containing a few errors (thus improving the PER) but it cannot correct packets containing a large number of errors, depending on the correction capability of the code. These highly corrupted and hence uncorrectable packets contribute to the high BER. This is illustrated in figure 5.3, where the histograms of the distribution of bits and bytes in error per packet are given for the 32 byte correction capability. The outer decoder can correct packets containing up to 32 erroneous bytes, and the PER is thus significantly reduced. However, some channels in a WLAN transmission lead to errors which are uncorrectable by the decoder. Packets containing more bytes in error than the correcting capability of the code can not be corrected and the number of bits in error therefore remains very high.

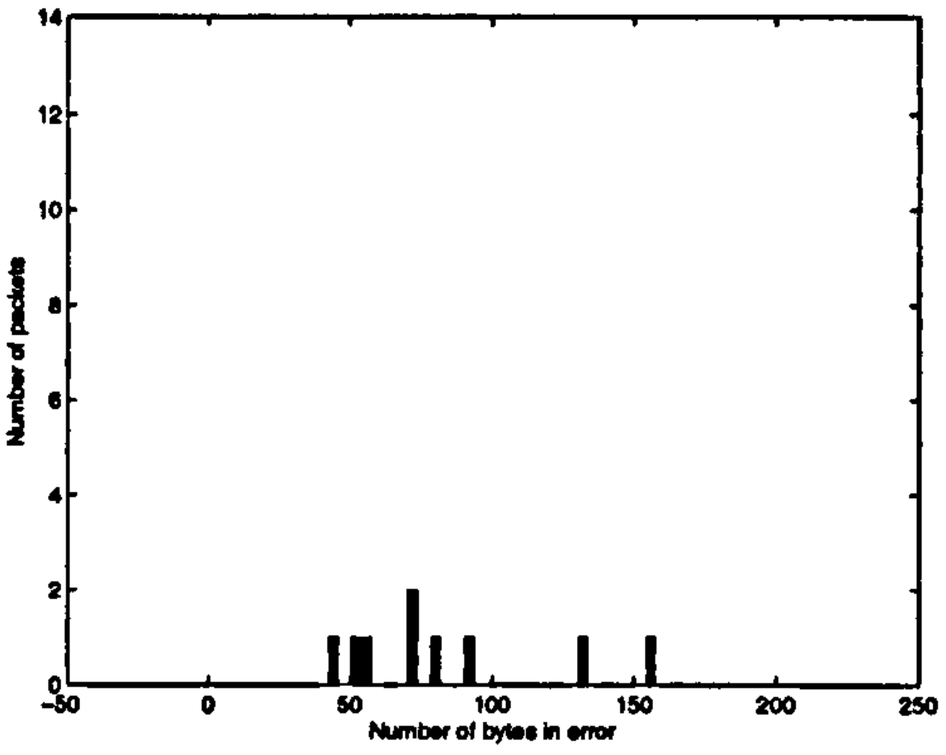
The DVB-T and DVB-S standards, which also employ an outer RS code with an OFDM based PHY layer, do not suffer as much as the home environment because they use directional antennas that guarantee near LOS channels, or at least channels with high K factors. This prevents deep fading and helps to ensure that errors are randomly distributed over packets, and not concentrated in a single and uncorrectable burst. To overcome highly corrupted packets, diversity techniques can be employed in combination with FEC.



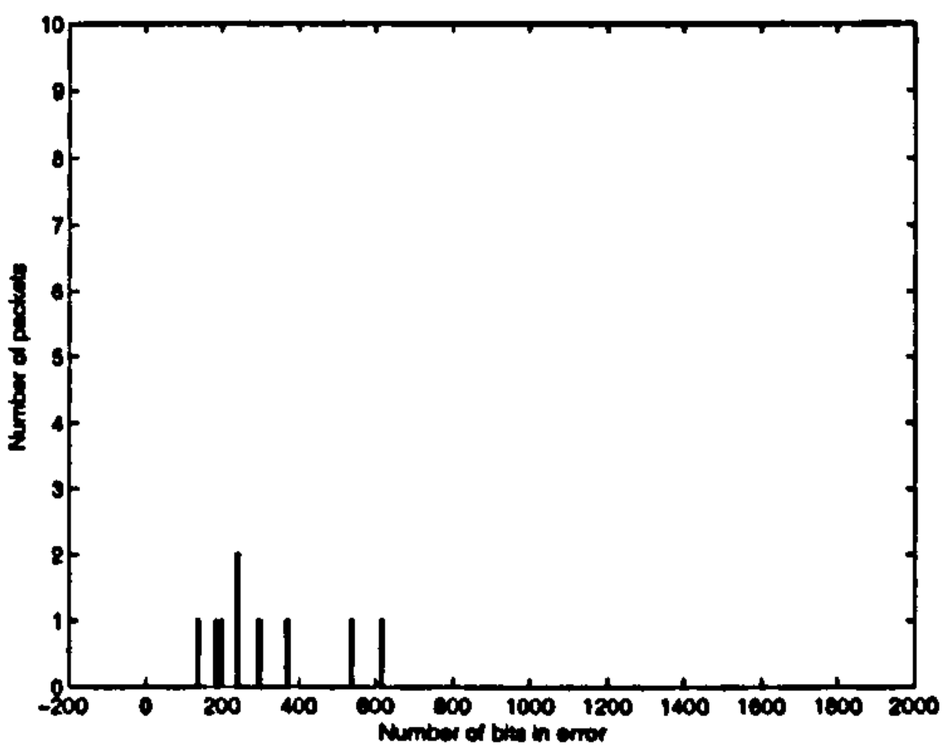
(a) Bytes in error, on the channel



(b) Bits in error, on the channel



(c) Bytes in error, After RS



(d) Bits in error, After RS

Figure 5.3: Error Distribution, with RS(188,252,t=32), mode 1, C/N=11dB, PDU size of 188 bytes

5.3 Space Time Block Codes

The radio performance of the WLAN PHY layer can be further enhanced using multiple transmit and receive antennas to provide spatial diverity. Alamouti in [142] derived a simple and attractive technique in the form of Space Time Block code (STBC), requiring only a small degree additional complexity [144]. This scheme was generalised by Tarokh in [143].

5.3.1 Mathematical Description of Space Time Block Codes

In this section, details are provided for the case of 2 transmitters and 1 receiver (2Tx-1Rx). Similar calculations have been derived for the 2Tx-2Rx case. At the receiver, after the FFT, the received signal on each of the 48 sub-carriers is:

$$y_k = H_k.x_k + n_k \tag{5.1}$$

where y_k is the received signal on the k^{th} sub-carrier, H_k represents the frequency response of the channel at the k^{th} sub-carrier and n_k represents the complex Additive

White Gaussian Noise (AWGN). x_k is the transmitted signal on the k^{th} subcarrier. In Alamouti's encoding scheme, two signals are transmitted simultaneously from two transmit antennas and received with one or two receive antennas. The transmission matrix is given by [142, 143]:

$$X = \begin{bmatrix} x_{1,k} & -x_{2,k}^* \\ x_{2,k} & x_{1,k}^* \end{bmatrix} \tag{5.2}$$

where $x_{1,k}$ and $x_{2,k}$ are the transmitted signals of two consecutive OFDM symbols respectively, at a given sub-carrier k, before being input to the IDFT and after the serial to parallel conversion (S/P).

At transmit antenna 1, for the k^{th} sub-carrier, $x_{1,k}$ is transmitted during the first symbol period followed by $-x_{2,k}^*$ in the second symbol period. At transmit antenna 2, $x_{2,k}$ is transmitted during the first symbol period followed by $x_{1,k}^*$ in the second symbol period. This is summarised in table 5.2. Each antenna transmits one OFDM symbol at half power compared to the standard IEEE 802.11a/g system.

Table 5.2: Encoding and Transmitting Sequence for the 2 Tx case

	Transmit Antenna 1	*Transmit Antenna 2*
First OFDM Period	$x_{1,k}$	$x_{2,k}$
Second OFDM Period	$-x_{2,k}^*$	$x_{1,k}^*$

At receive antenna 1, after the DFT and the removal of the cyclical prefix, the received signal is given by [142]:

$$\begin{bmatrix} y_{1,k} \\ y_{2,k} \end{bmatrix} = \begin{bmatrix} H_{1,k} \\ H_{2,k} \end{bmatrix} \cdot \begin{bmatrix} x_{1,k} & -x_{2,k}^* \\ x_{2,k} & x_{1,k}^* \end{bmatrix} + \begin{bmatrix} n_{1,k} \\ n_{2,k} \end{bmatrix} \tag{5.3}$$

where $y_{1,k}$ and $y_{2,k}$ represent the received signals during the first and second symbol period respectively. $n_{1,k}$ and $n_{2,k}$ represent the AWGN and $H_{1,k}$ and $H_{2,k}$ represents the frequency responses for sub-carrier k of the channels between Tx 1 and Rx 1 and between Tx 2 and Rx 1 respectively. Equation 5.3 can be rewritten as [142, 143]:

$$y_{1,k} = \quad x_{1,k}.H_{1,k} + x_{2,k}.H_{2,k} + n_{1,k} \tag{5.4}$$

$$y_{2,k} = -x_{2,k}^*.H_{1,k} + x_{1,k}^*.H_{2,k} + n_{2,k} \tag{5.5}$$

It is assumed that the channel responses are uncorrelated and remain constant during the period of two OFDM symbols. IEEE and ETSI OFDM based WLAN use an ODFM period of $4\mu s$, this assumption is reasonable for an indoor system. After channel estimation, channel parameters are known to the receiver and according to

[144] and [145], $y_{1,k}$ and $y_{2,k}$ can be combined into:

$$s_{1,k} = y_{1,k}.H^*_{1,k} + y^*_{2,k}.H_{2,k} \tag{5.6}$$

$$s_{2,k} = y_{1,k}.H^*_{2,k} - y^*_{2,k}.H_{1,k} \tag{5.7}$$

By using equations (5.4) and (5.5), (5.6) and (5.7) are rewritten as:

$$s_{1,k} = x_{1,k}(|H_{1,k}|^2 + |H_{2,k}|^2) + n_{1,k}.H^*_{1,k} + n^*_{2,k}.H_{2,k} \tag{5.8}$$

$$s_{1,k} = x_{2,k}(|H_{1,k}|^2 + |H_{2,k}|^2) + n_{1,k}.H^*_{2,k} - n^*_{2,k}.H_{1,k} \tag{5.9}$$

In order to perform Soft Decision Viterbi decoding, the Channel State Information (CSI) of both channels and for all the sub-carriers ($H_{1,k}$,$H_{2,k}$) is passed to the decoder in order to calculate the metric. Figure 5.4 describes the block diagram for a 2Tx-1Rx STBC configuration that will be used as the inner part of the combined system.

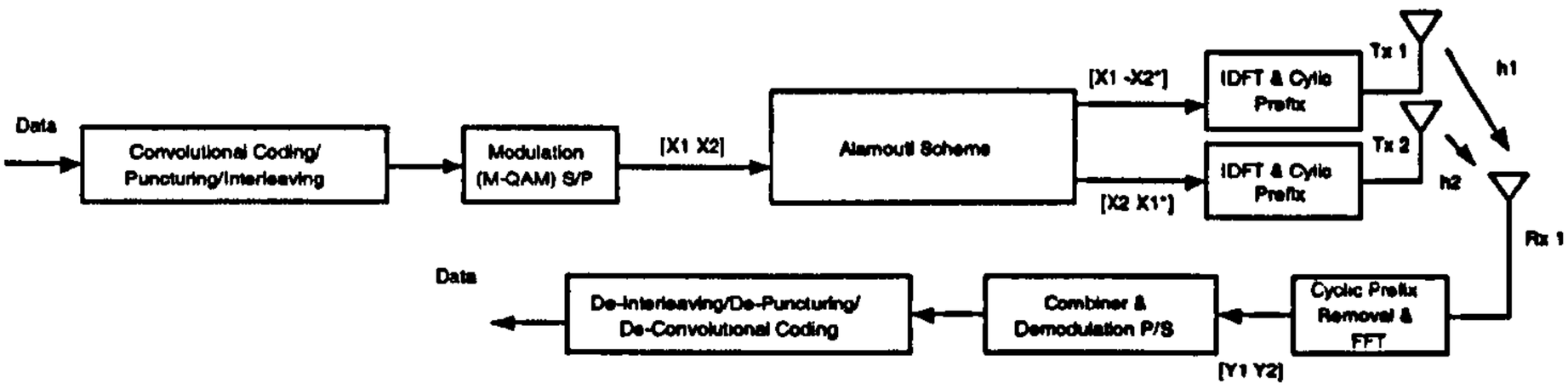
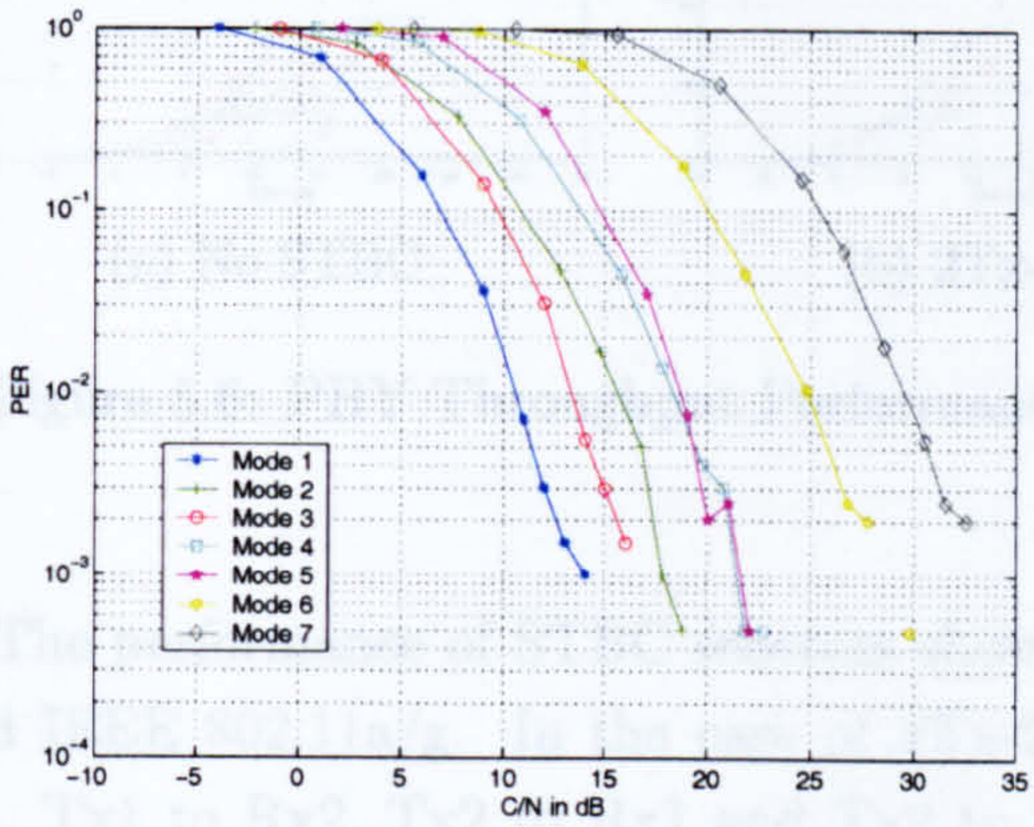


Figure 5.4: 2Tx-1Rx STBC block diagram for IEEE802.11a PHY layer

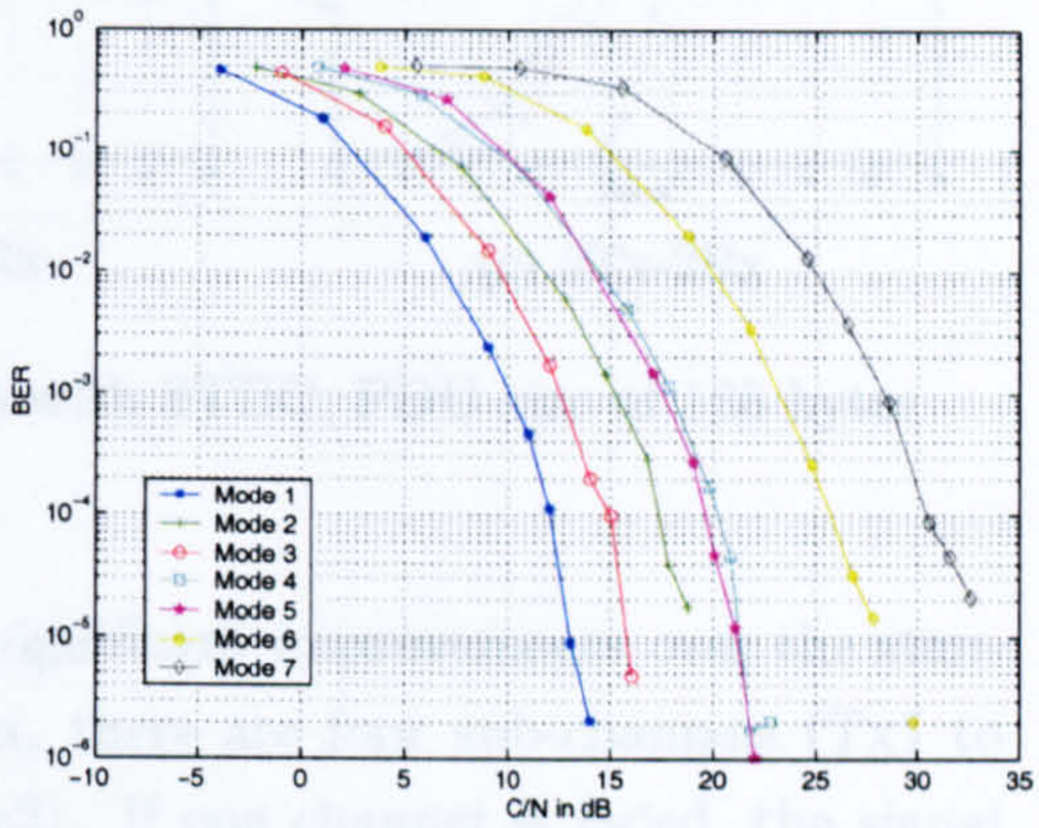
5.3.2 Performance Results

In [144], the authors showed that a 2Tx-2Rx system can achieve a gain of 8.5 dB for mode 1 compared to a standard 1Tx-1Rx system over IEEE802.11a for a PDU size of 54 bytes. Since, MPEG-2 TS packets are used here, in this section performance results with a PDU size of 188 bytes are presented. Figure 5.5 shows the PER and BER performance versus C/N for the IEEE 802.11a standard PHY layer, the 2Tx-1Rx case and the 2Tx-2Rx case. It can be observed that all the modes are improved, and the PER and BER are reduced for a given C/N. The throughput enhancements at the PHY are shown in figure 5.6.

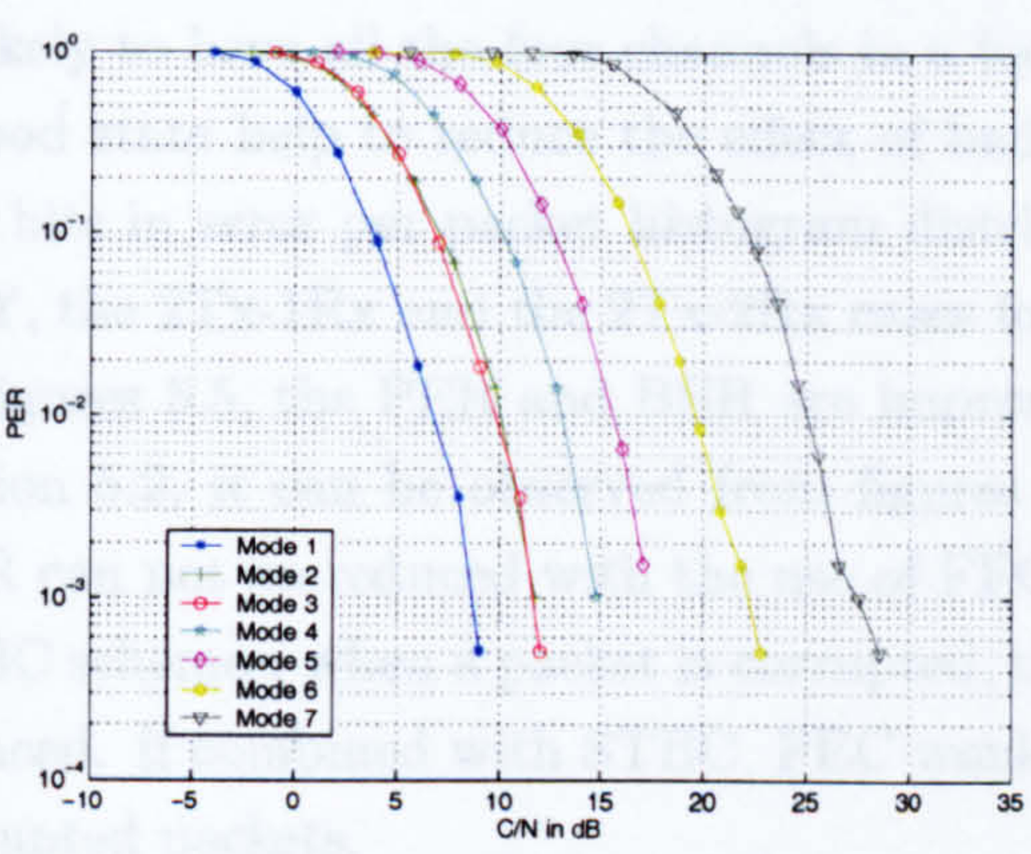
The BER, PER and throughput enhancements for mode 3 are shown in figure 5.7 and summarised in table 5.3 for the 2Tx-1Rx and 2Tx-2Rx cases and for a packet size of 188 bytes. Significant improvements can be observed and a gain of 5dB is achieved by using 2 transmitters and one receiver over the legacy IEEE 802.11a/g PHY layer for a BER of 10^{-4}. A further 5dB can be gained with two receivers. For a throughput of 10Mbits/s with mode 3, two transmitters allow a gain of 6dB over the IEEE 802.11a/g PHY layer. Table 5.4 details the improvements for all modes for a PER of 10^{-2} for the 2Tx-2Rx case. Observed gains are from 8dB for modes 1 and 2 up to 10dB for mode 2, 6, 7.



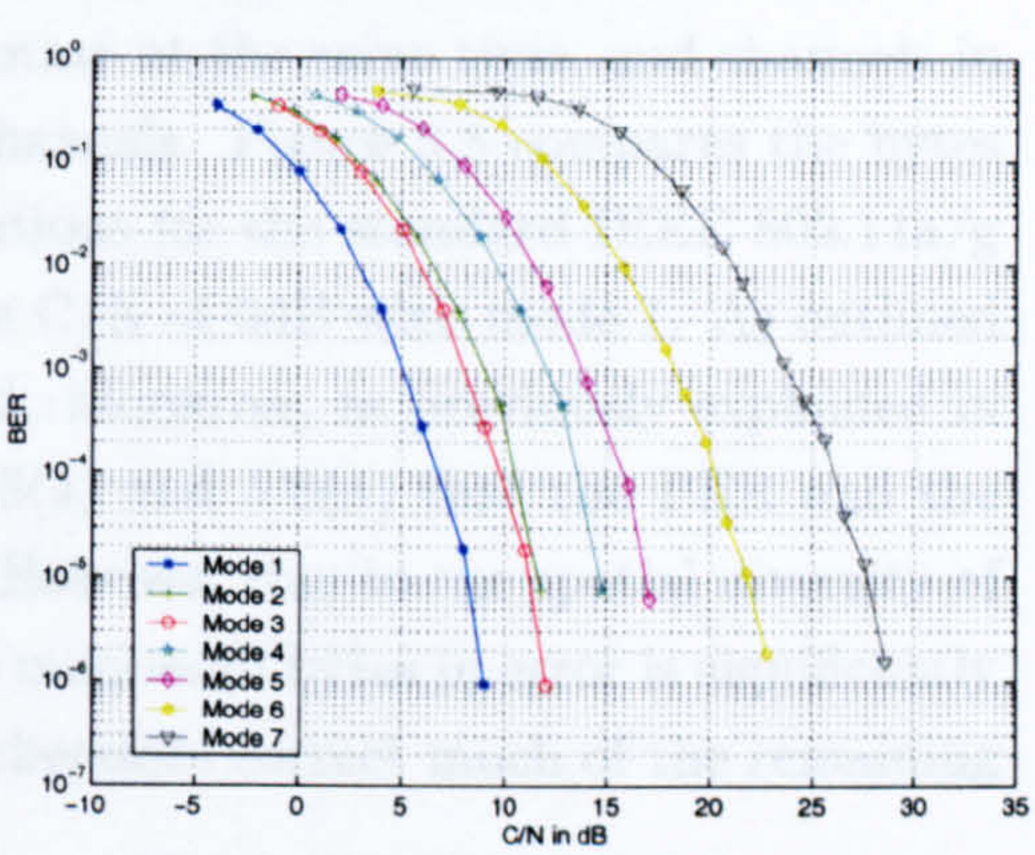
(a) PER - No STBC



(b) BER - No STBC



(c) PER - 2Tx-1Rx

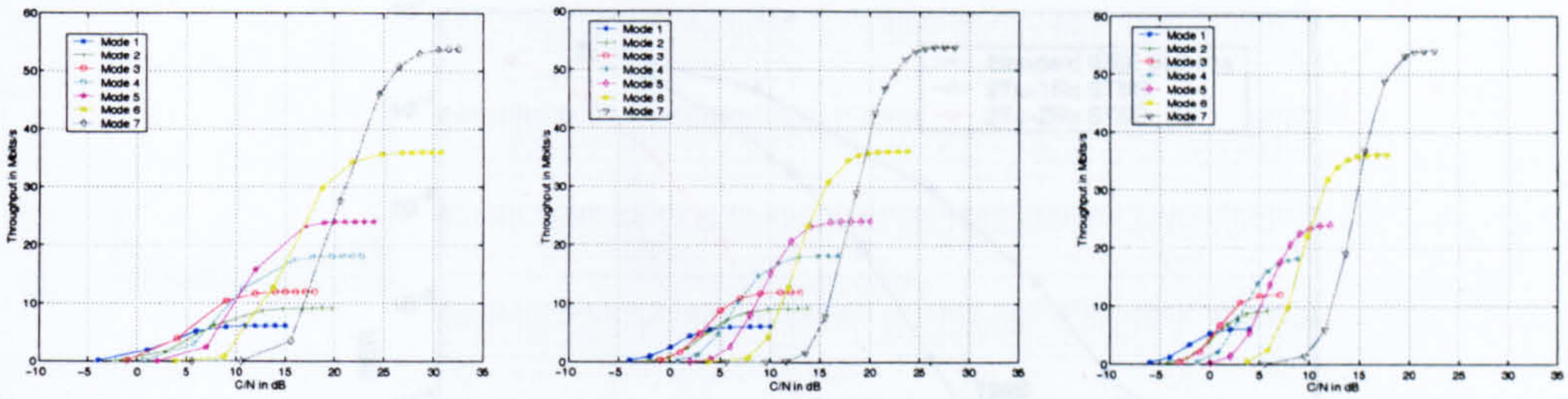


(d) BER - 2Tx-1Rx

(e) PER - 2Tx-2Rx

(f) BER - 2Tx-2Rx

Figure 5.5: Performances with STBC, PDU size of 188 bytes



(a) No STBC (b) 2Tx-1Rx (c) 2Tx-2Rx

Figure 5.6: PHY Throughput Performances with STBC, PDU size of 188 bytes

The performance of STBC schemes shows significant improvements over the standard IEEE 802.11a/g. In the case of 2Tx-2Rx, there are four sub-channels (Tx1 to Rx1, Tx1 to Rx2, Tx2 to Rx1 and Tx2 to Rx2). If one channel is faded, the signal is not necessarily corrupted, depending on the state of the other three channels. It is unlikely to have all the four channels in a bad state at the same time, and channels in a good state help to reduce the effect of bad channels. Figure 5.8 compares the bytes and bits in error per packet histogram distributions for the standard IEEE 802.11a/g PHY, the 2Tx-1Rx and the 2Tx-2Rx cases for a C/N of 0dB with mode 1. As outlined by figures 5.5, the PER and BER are improved. Moreover, as previously explained in section 5.2, it can be observed from figures 5.8(a) and 5.8(d) that the PER and the BER can not be reduced with the use of FEC. However, due to the spatial diversity of STBC schemes, when a packet is corrupted, the number of bytes in error is significantly reduced. If combined with STBC, FEC would therefore correct much of the remaining corrupted packets.

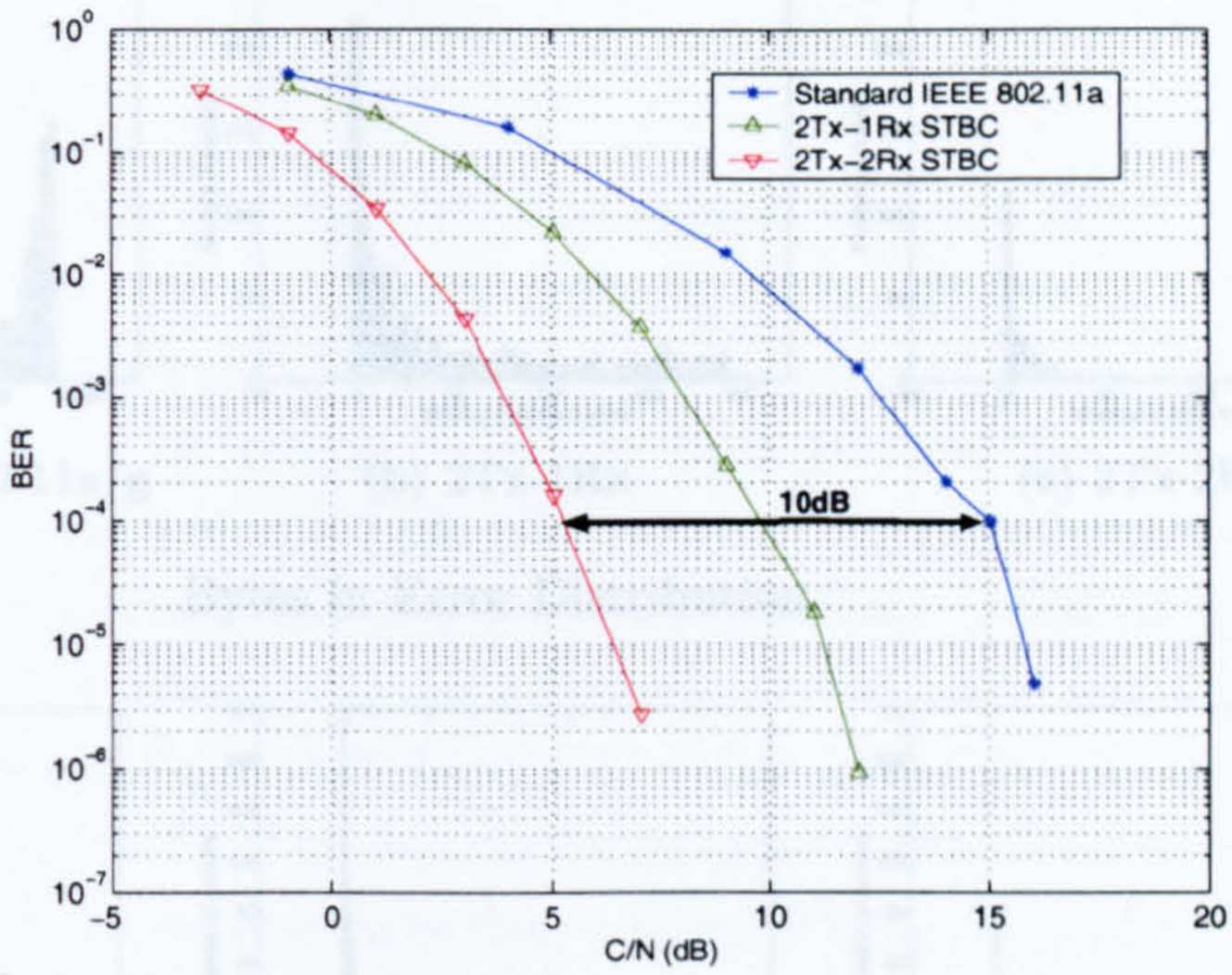
Table 5.3: STBC Improvements over IEEE 802.11a/g, for mode 3

	2Tx-1Rx	*2Tx-2Rx*
$PER = 10^{-2}$	3.5 dB	8 dB
$BER = 10^{-4}$	5 dB	10 dB
Throughput=10Mbits/s	2.7 dB	6 dB

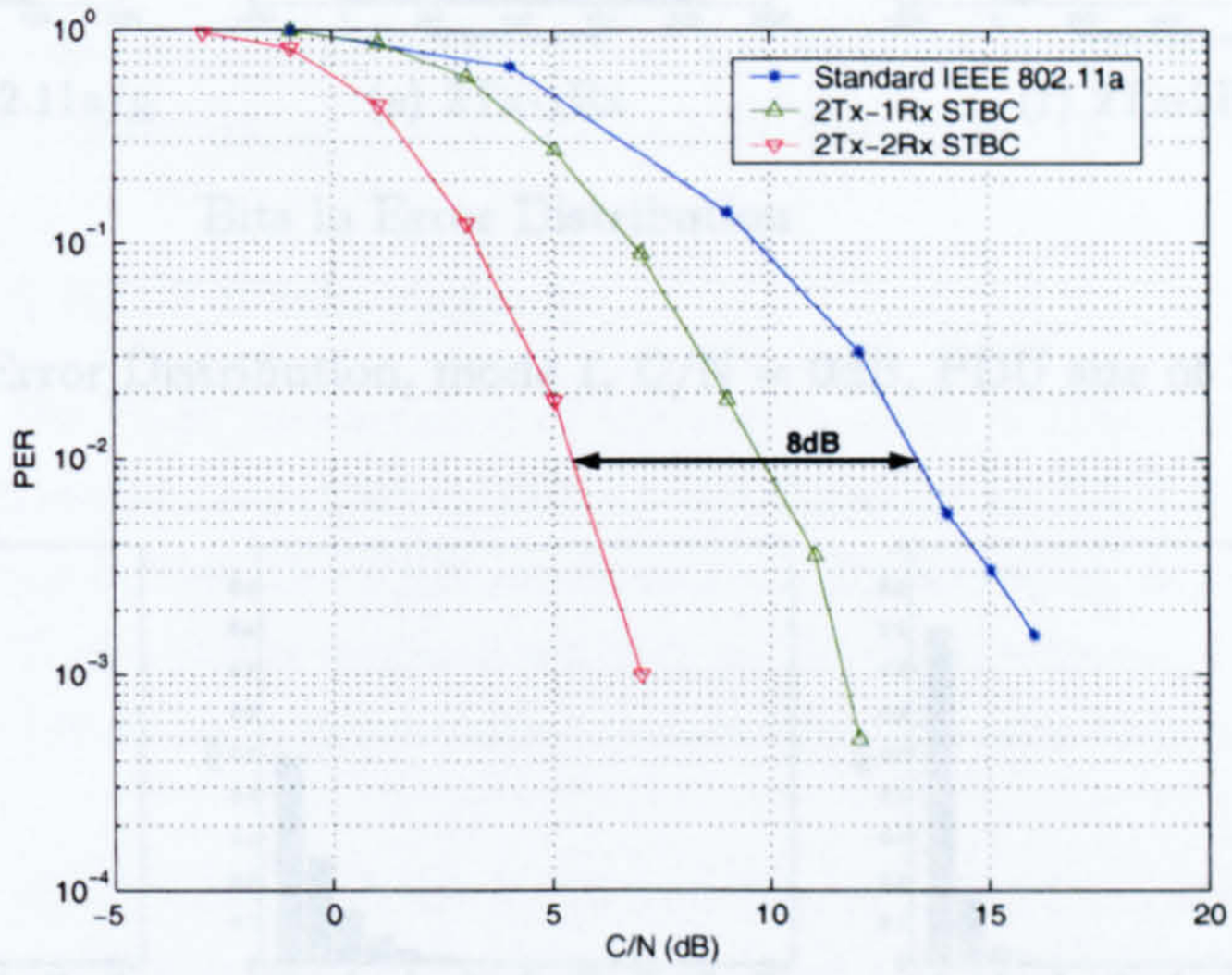
Table 5.4: 2Tx-2Rx STBC gain at PER=1%

Mode	1	2	3	4	5	6	7
Gain in dB	8	10	8	10	8.5	10	10

Figure 5.9 shows the improvements of STBC in terms of use of retransmission and delay. It can be seen that a 2Tx-1Rx reduces the number of ARQ, and the use of 2Tx-2Rx reduces even further the number of retransmission. The delay introduced are thus also minised.



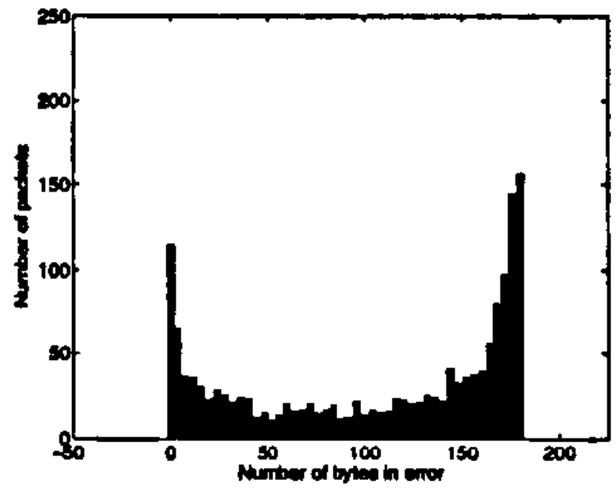
(a) BER Improvements



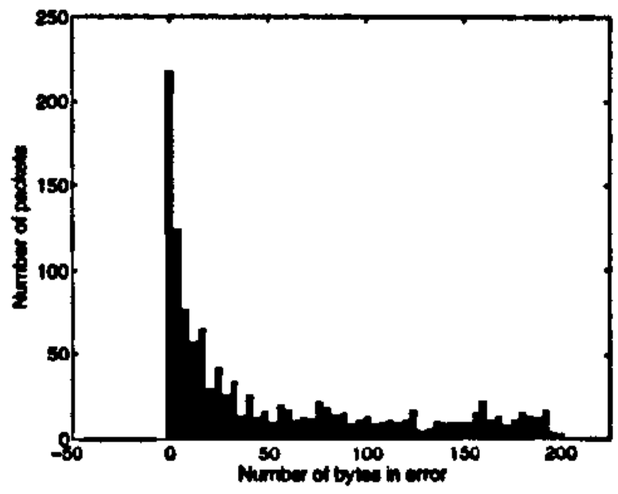
(b) PER Improvements

(c) PHY Throughput Improvements

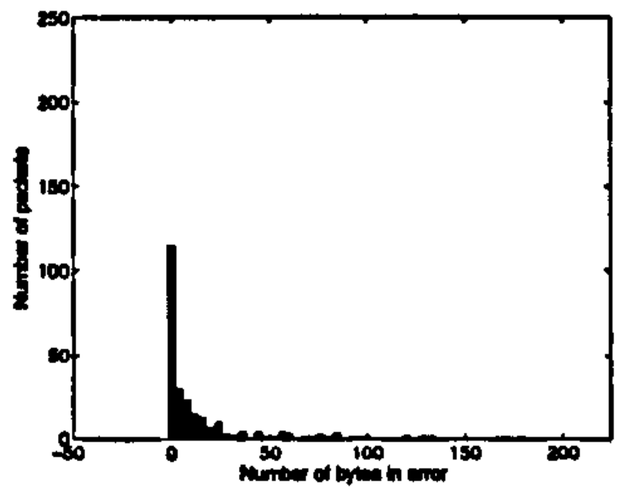
Figure 5.7: Improvement of STBC over IEEE 802.11a/g, for mode 3, PDU size of 188 bytes



(a) Standard IEEE802.11a/g

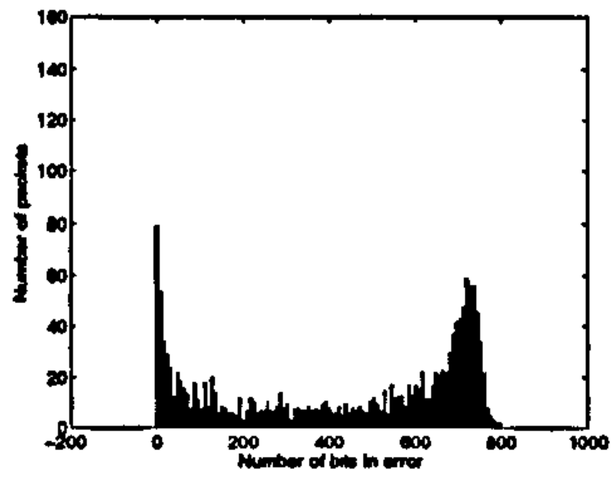


(b) 2Tx-1Rx

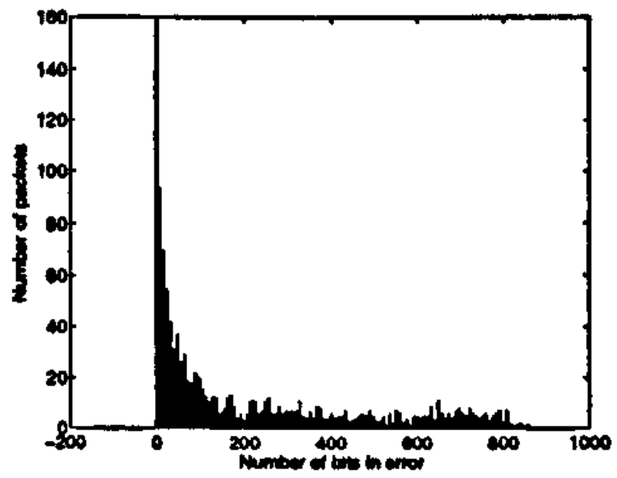


(c) 2Tx-2Rx

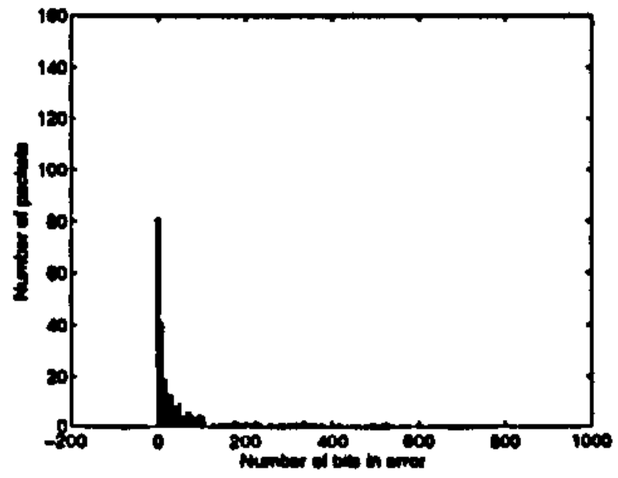
Bytes in Error Distribution



(d) Standard IEEE802.11a/g



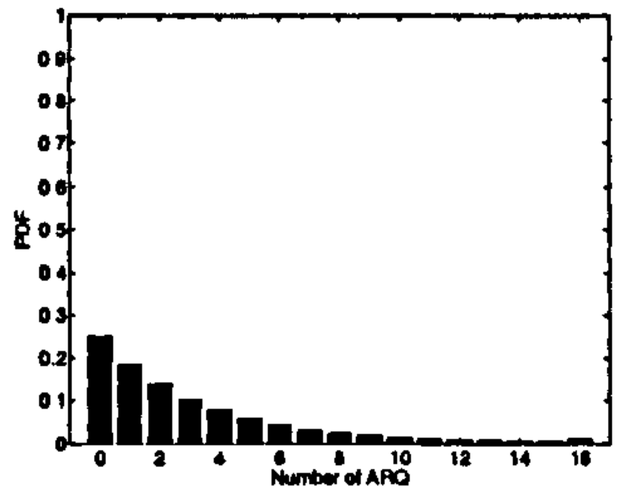
(e) 2Tx-1Rx



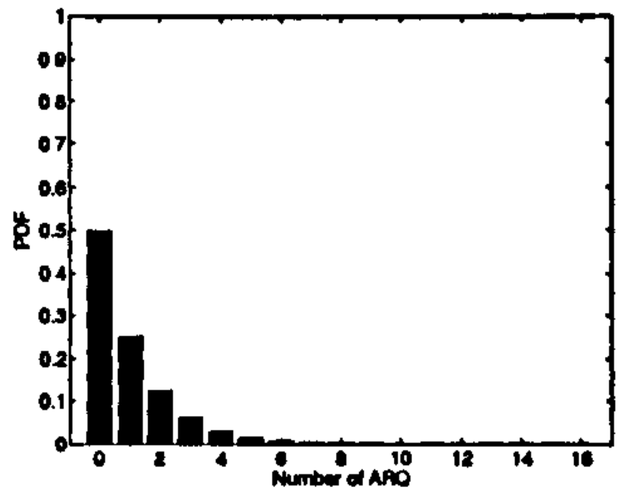
(f) 2Tx-2Rx

Bits in Error Distribution

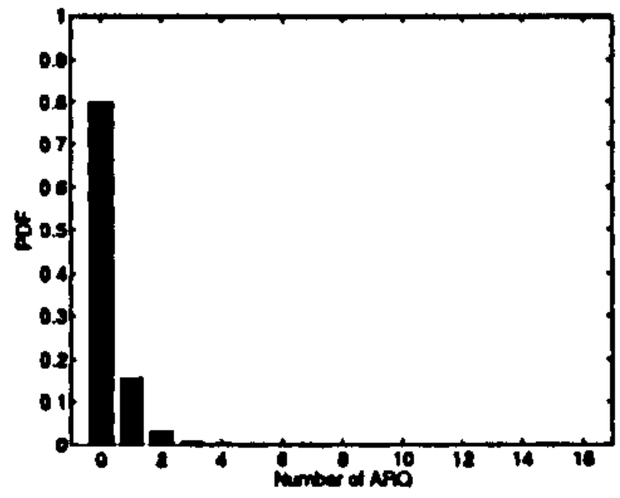
Figure 5.8: STBC Error Distribution, mode 1, C/N = 0dB, PDU size of 188 bytes



(a) Standard IEEE802.11a/g

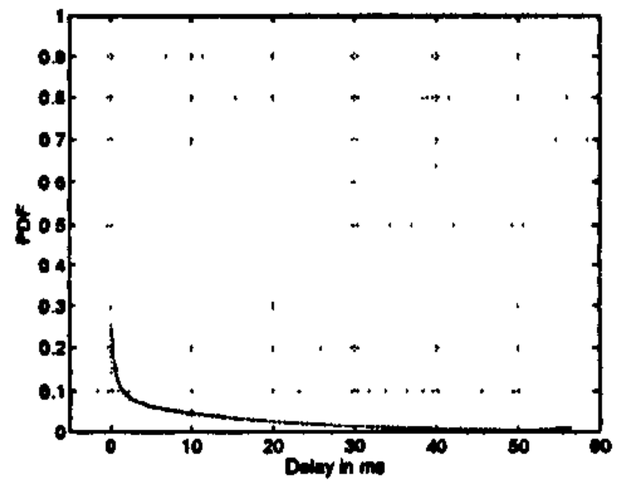


(b) 2Tx-1Rx

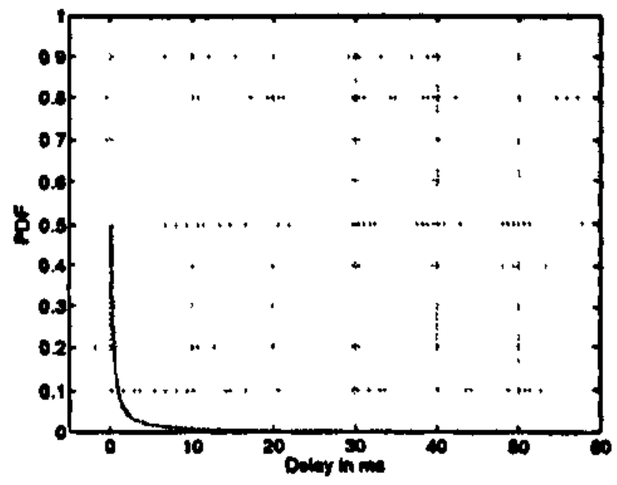


(c) 2Tx-2Rx

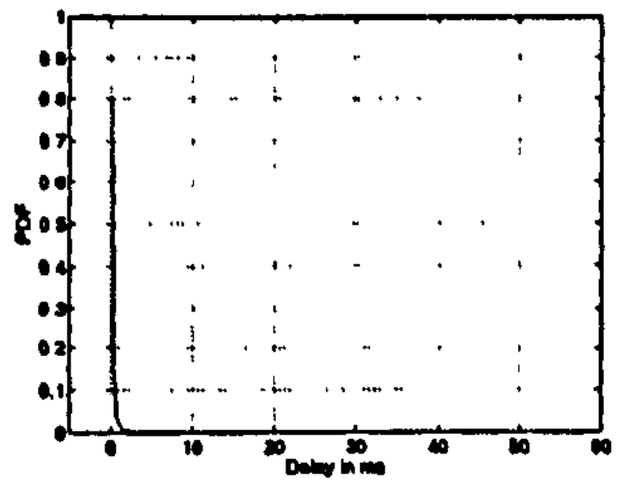
PDF ARQ



(d) Standard IEEE802.11a/g



(e) 2Tx-1Rx



(f) 2Tx-2Rx

PDF Delay

Figure 5.9: ARQ and delay Performance of STBC for mode 3 - C/N = 2.5dB

5.4 Combined FEC and STBC

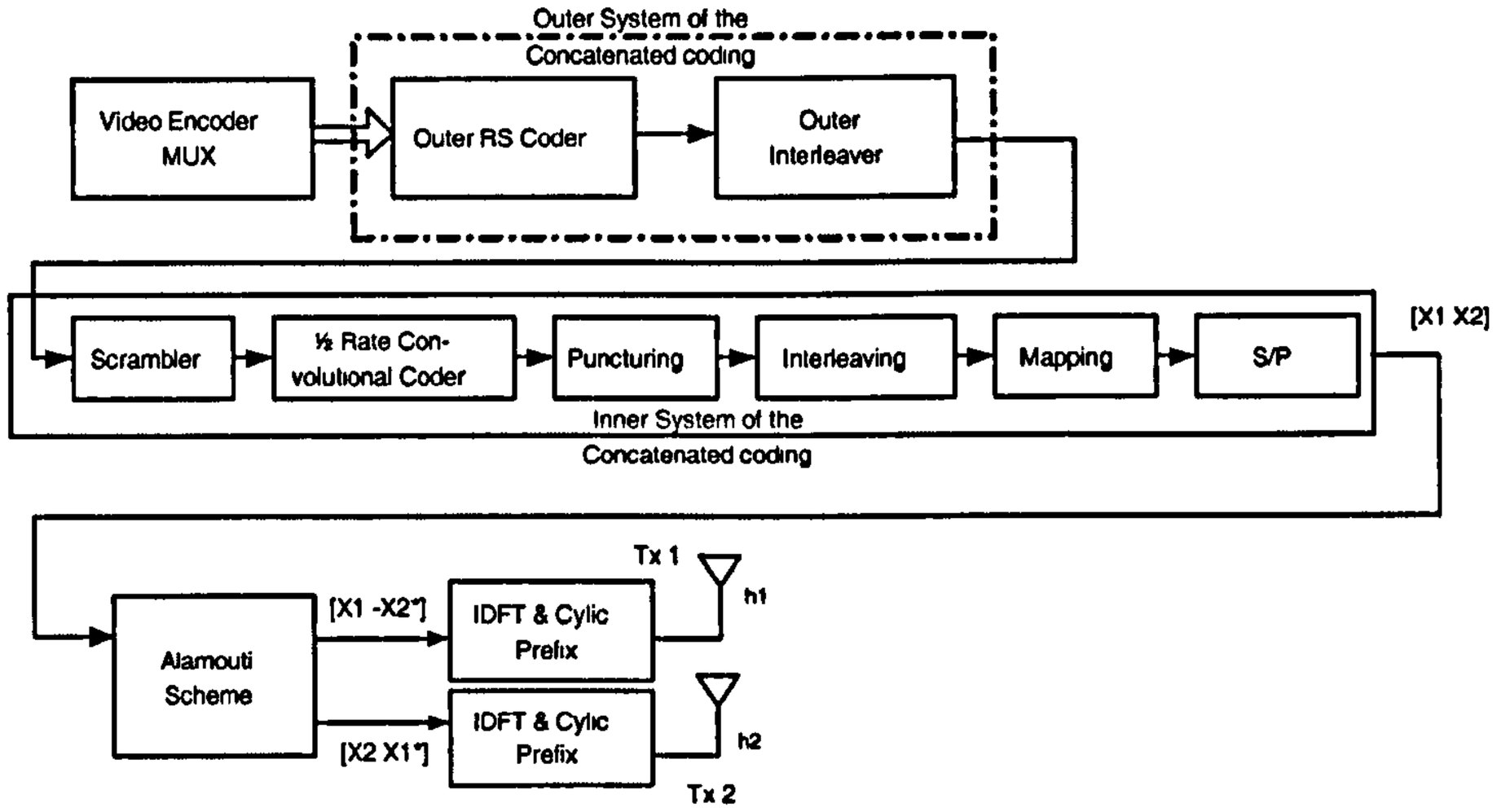
5.4.1 Overview

As shown in section 5.2, a simple RS code applied with the IEEE 802.11a/g PHY layer does not alone provide an efficient means to reduce the use of ARQ. The STBC spatial diversity technique allows reduction of the number of highly corrupted packets that are uncorrectable. A system combining both systems (RS and STBC) would improve further the performance [138]. The proposed system is therefore composed of an outer RS code with two different correction capabilities (8 and 32 bytes) and with an outer interleaver combined with an STBC system (2Tx-1Rx and 2Tx-2Rx) at the PHY layer, as shown in figure 5.10 for the 2Tx-1Rx case. The convolutional encoder and the interleaver of the PHY layer form the inner part of the concatenated code. The reference system for the simulation is the standard IEEE 802.11a PHY layer without the RS encoder and without STBC. TS packets are 188 byte long and are input to the outer RS encoder. The encoded packets, with a length of 204 bytes or 252 bytes depending on the correction capabilities chosen, are then input to the modified IEEE 802.11a PHY layer implementing a 2Tx-1Rx or 2Tx-2Rx STBC (see Table 5.1).

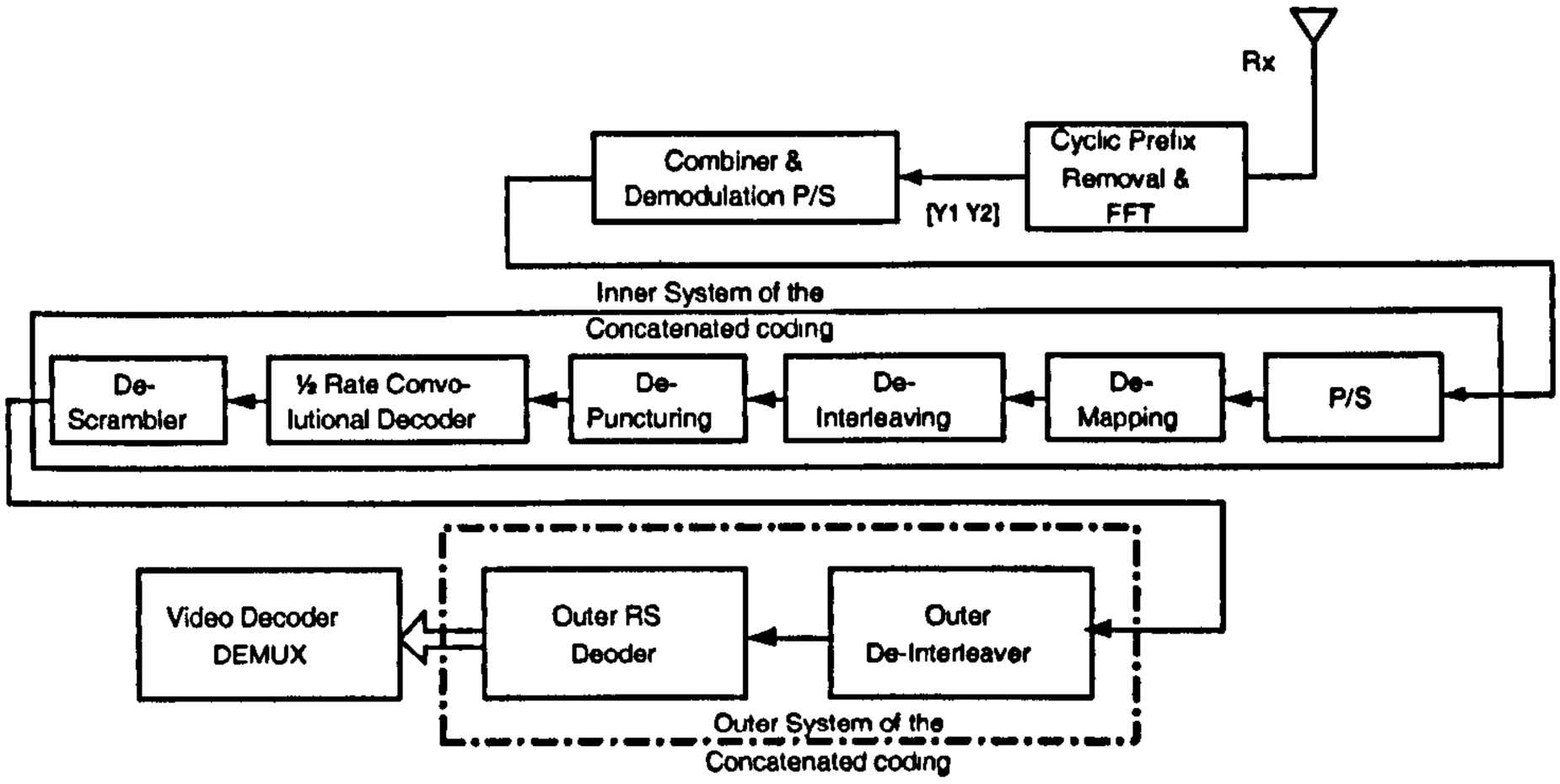
5.4.2 PER and BER Performance

Figure 5.11 shows the PER performance of the system with a 2Tx-2Rx configuration for the different correction capabilities and for the 7 modes of IEEE 802.11a/g. The RS encoder improves further the PER performance. For example, at a PER of 10^{-2} using mode 5, a C/N of 11dB is required without an RS code, whereas, it only requires 9.5dB and 8dB for RS codes with correction capabilities of 8 and 32 bytes respectively [138].

The histogram distributions of the bytes and bits in error for mode 1 are shown in figure 5.12 and this illustrates the improvement over the standard IEEE 802.11a/g PHY layer with STBC 2Tx-2Rx and RS(188,204,t=8). The number of highly corrupted packets is greatly reduced with STBC, hence the RS code can further correct packets with only a few errors. However, the few packets with a large number of bits in error are still uncorrectable and still drive the BER. Figure 5.13 shows the PER and BER performance respectively for the combined system for mode 3 with the 2Tx-1Rx and 2Tx-2Rx STBC configurations and for the two correction capabilities. The BER is slightly improved. The 2Tx-1Rx and the 2Tx-2Rx STBC configurations without RS encoding provide an improvement of 3.5dB and 8dB respectively over the IEEE 802.11a/g standard for a PER of 10^{-2}. The proposed combined system using a 2Tx-1Rx configuration offers a further 1.5dB and 2.5dB gain with 8 and 32 byte correction capabilities respectively, leading therefore to a 5dB and 6dB gain over the standard [138]. The 2Tx-2Rx configuration offers 8.5dB and 10dB gains over the standard with the 8 and 32 byte correction capabilities respectively. For a given C/N, the stronger RS code with a 2Tx-2Rx configuration will reduce considerably the PER and reduce therefore the undesirable use of ARQ for video time-bounded transmissions [138]. PER

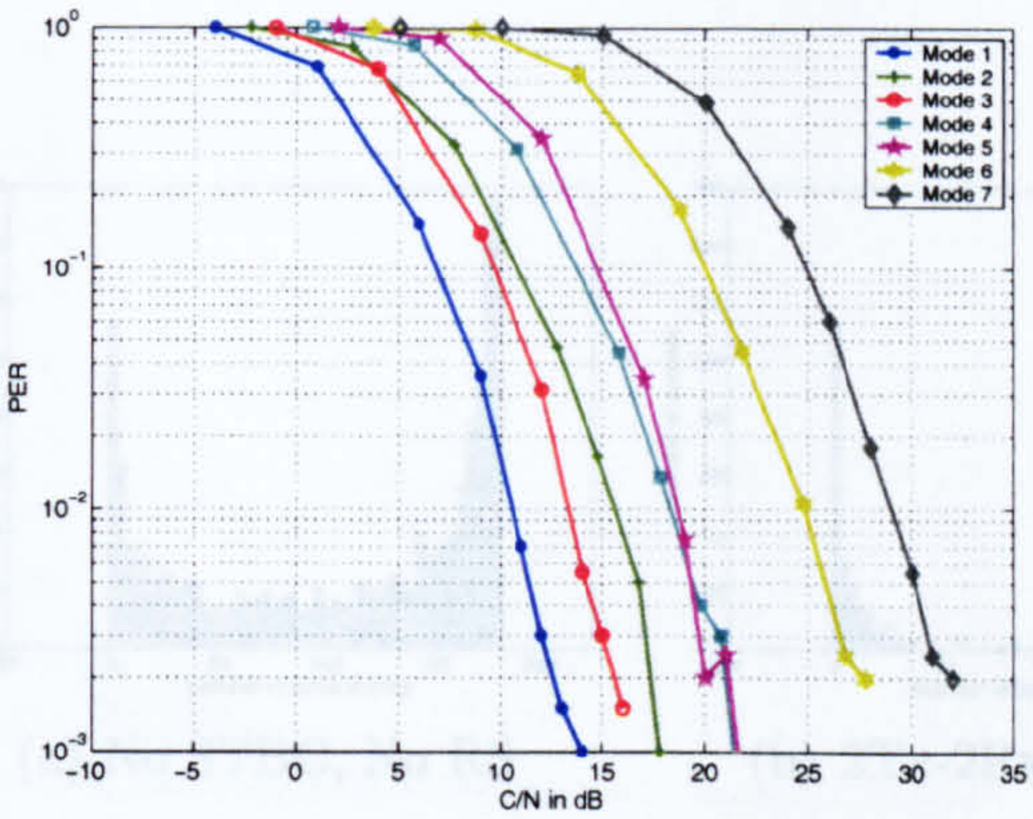


(a) Transmitter

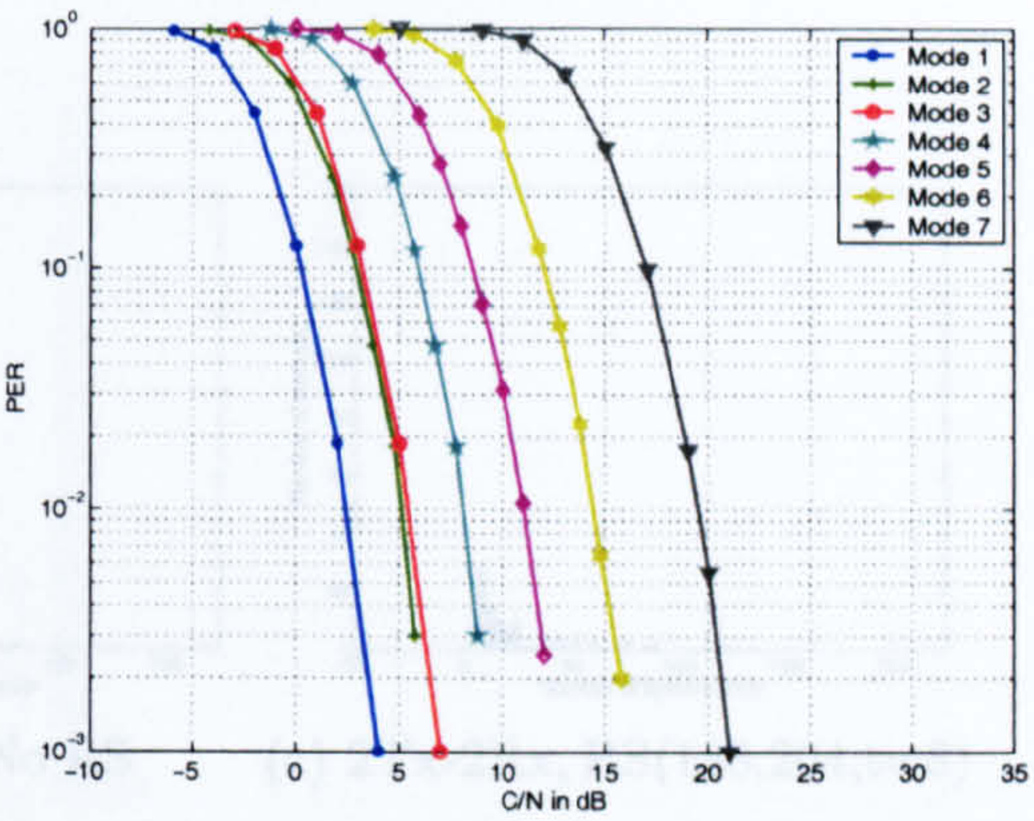


(b) Receiver

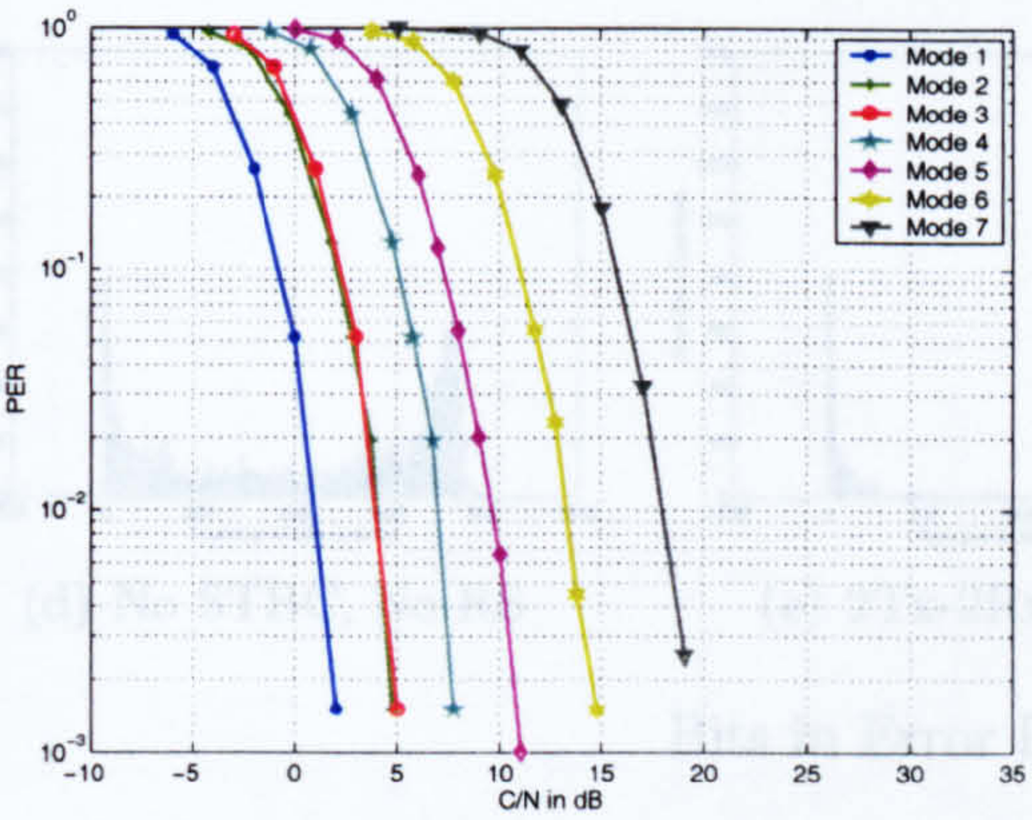
Figure 5.10: Block Diagram of the Combined Concatenated FEC coding and STBC: 2Tx-1Rx



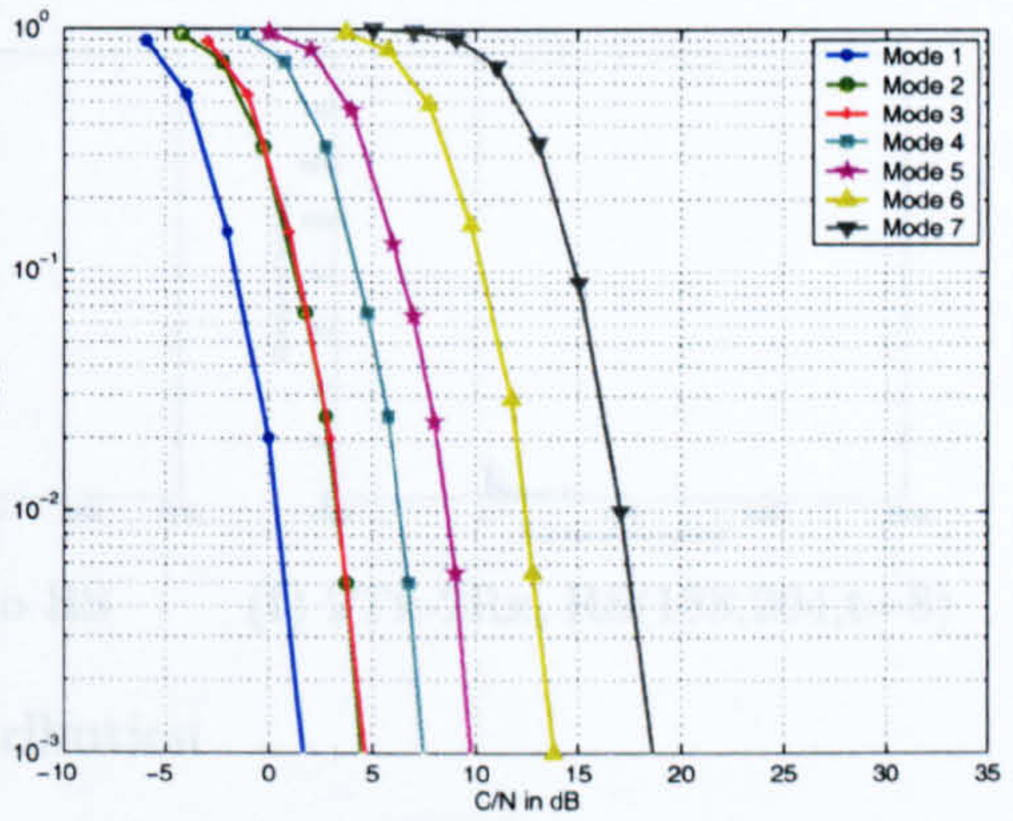
(a) no RS, no STBC



(b) no RS, 2Tx-2Rx



(c) RS(188,204,t=8), 2Tx-2Rx



(d) RS(188,252,t=32), 2Tx-2Rx

Figure 5.11: PER performances for a 2Tx-2Rx STBC and different correction capabilites

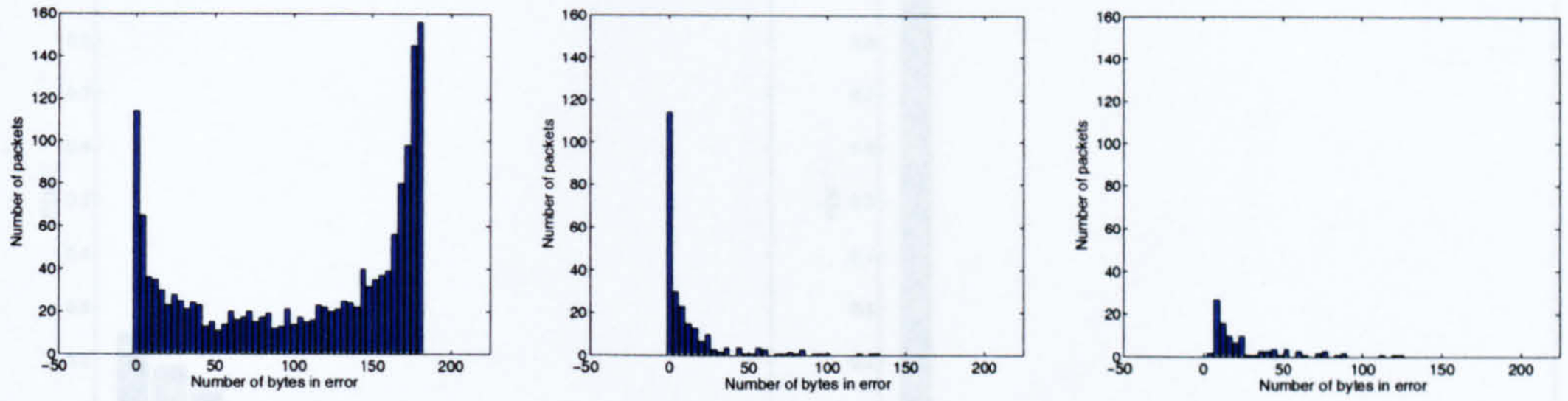
improvements are summarised in table 5.5.

Table 5.5: Gain at PER=1% with Mode 3

Mode	2-1, No RS	2-1, t=8	2-1, t=32	2-2, No RS	2-2, t=8	2-2, t=32
Gain in dB	3.5	5	6	8	9.5	10

5.4.3 ARQ and Delay Performance

Figure 5.14 shows the probability density functions of the number of retransmissions and delays for mode 3 for a C/N of 2.5dB for the standard IEEE 802.11a compared to the combination of a 2Tx-2Rx STBC with RS(188,252,t=32). It can be seen that the proposed system considerably reduces the use of ARQ. 0 ARQ corresponds to the first transmission. Thanks to a reduced PER, packets are more likely to be received correctly after the first transmission with the proposed system and the probability of using more than one ARQ is null with a C/N of 2.5dB. The delays are therefore considerably reduced as shown in figures 5.14(c) and 5.14(d). The proposed combined system constrains the delay under 1 ms whereas the standard IEEE 802.11a allows



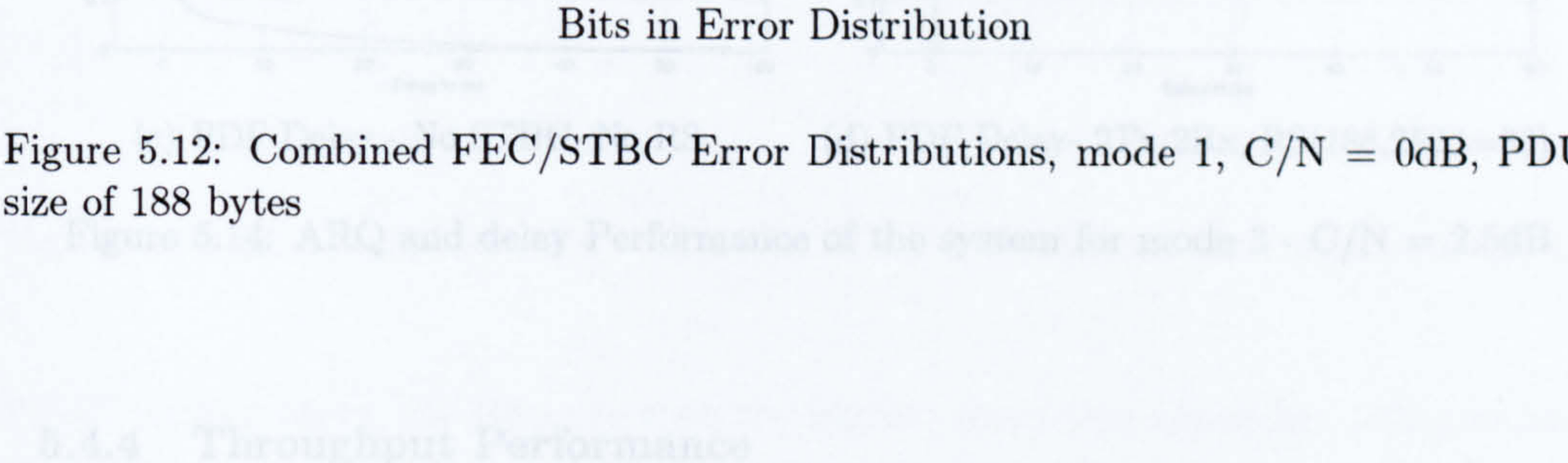
(a) No STBC, No RS (b) 2Tx-2Rx, No RS (c) 2Tx-2Rx, RS(188,204,t=8)

Bytes in Error Distribution

(d) No STBC, No RS (e) 2Tx-2Rx, No RS (f) 2Tx-2Rx, RS(188,204,t=8)

Bits in Error Distribution

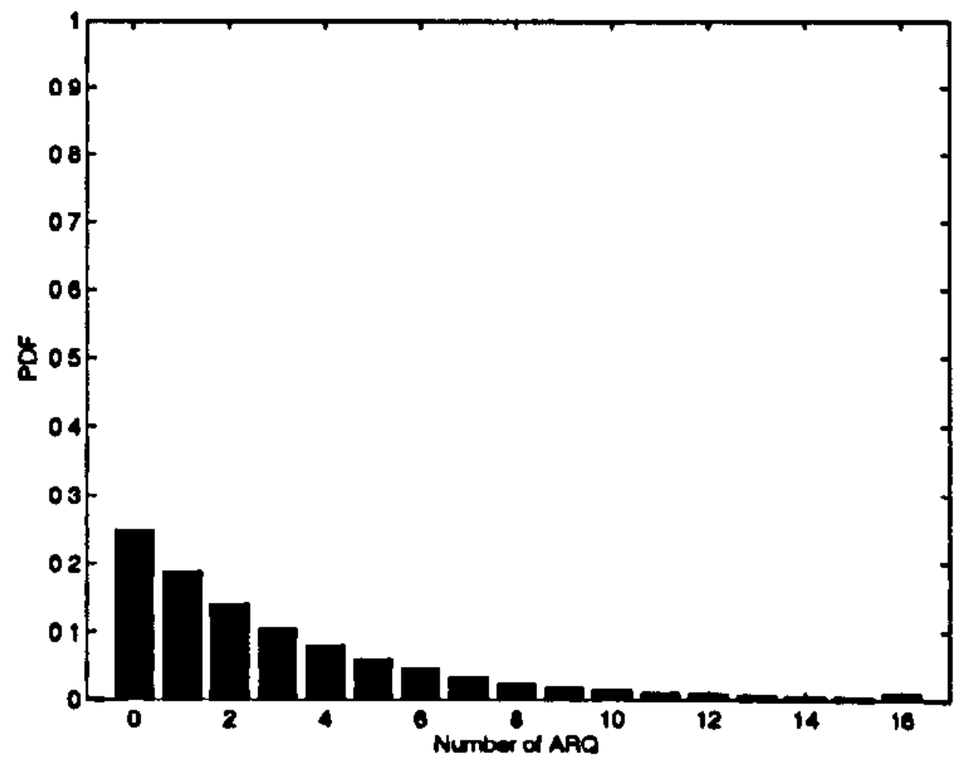
Figure 5.12: Combined FEC/STBC Error Distributions, mode 1, C/N = 0dB, PDU size of 188 bytes



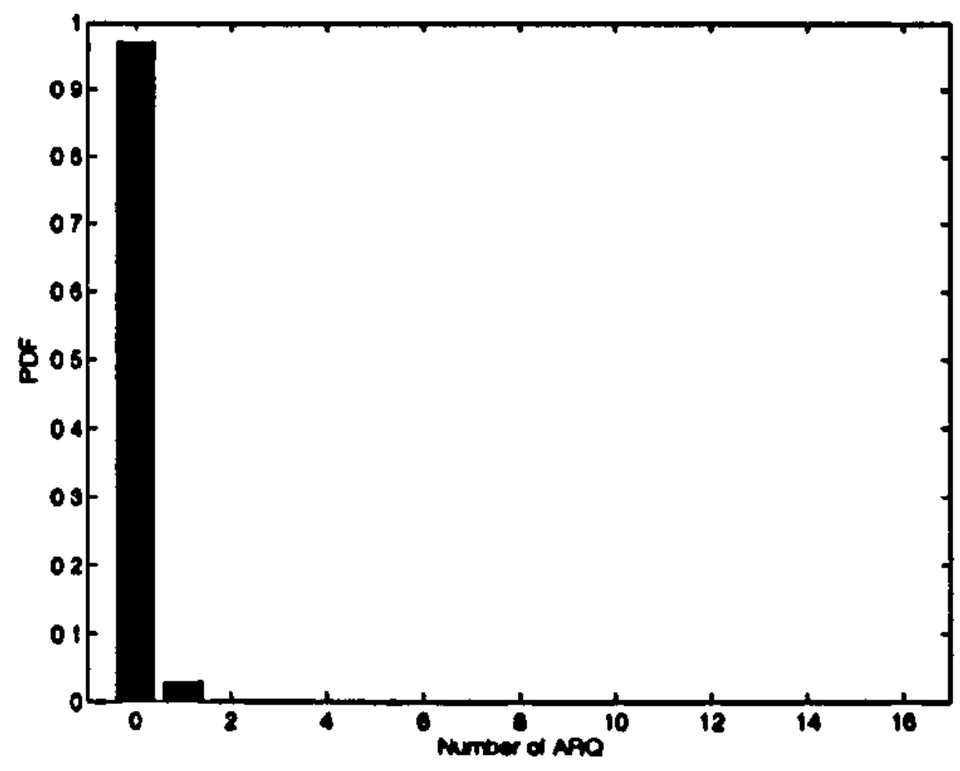
(a) PER Performance (b) BER Performance

Figure 5.13: Performance of the system for mode 3 - Channel A

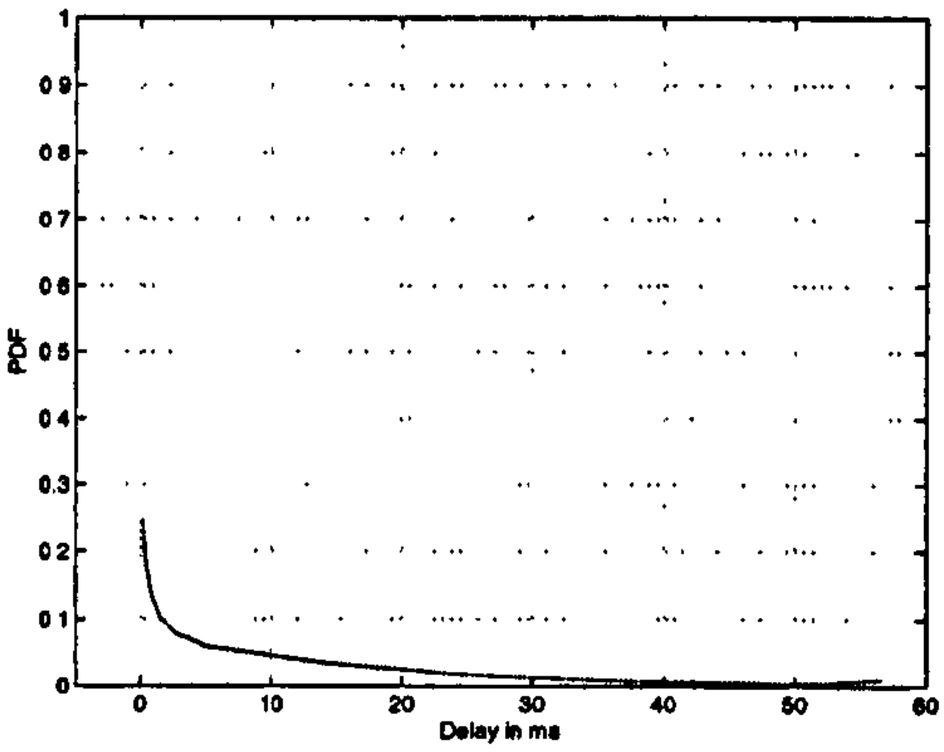
delay up 40 ms.



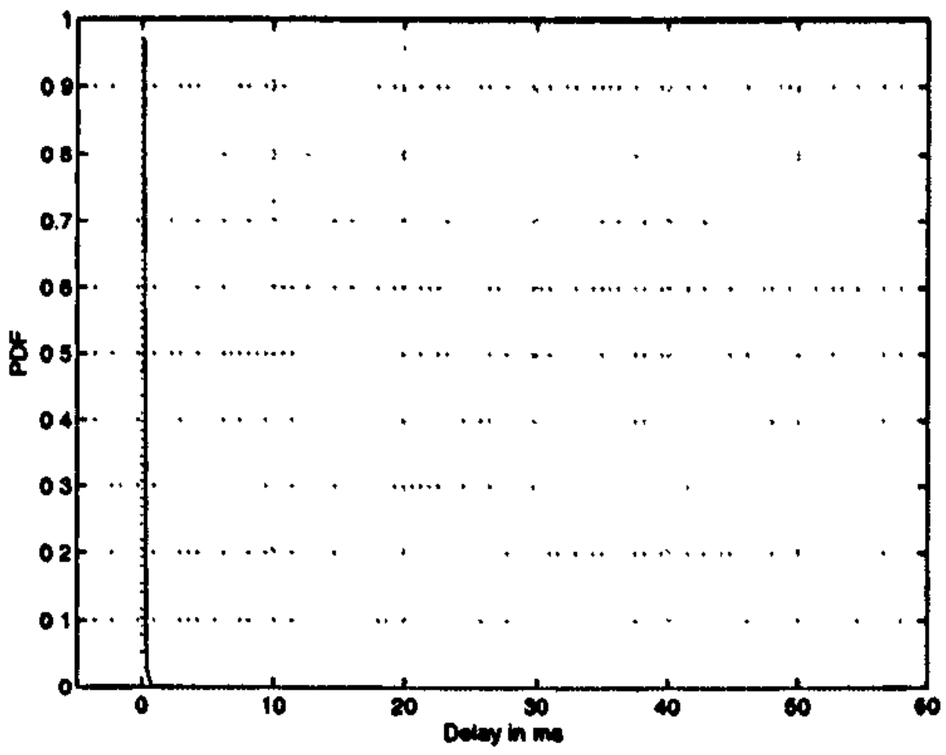
(a) PDF ARQ - No STBC, No RS



(b) PDF ARQ - 2Tx-2Rx, RS(188,252,t=32)



(c) PDF Delay - No STBC, No RS



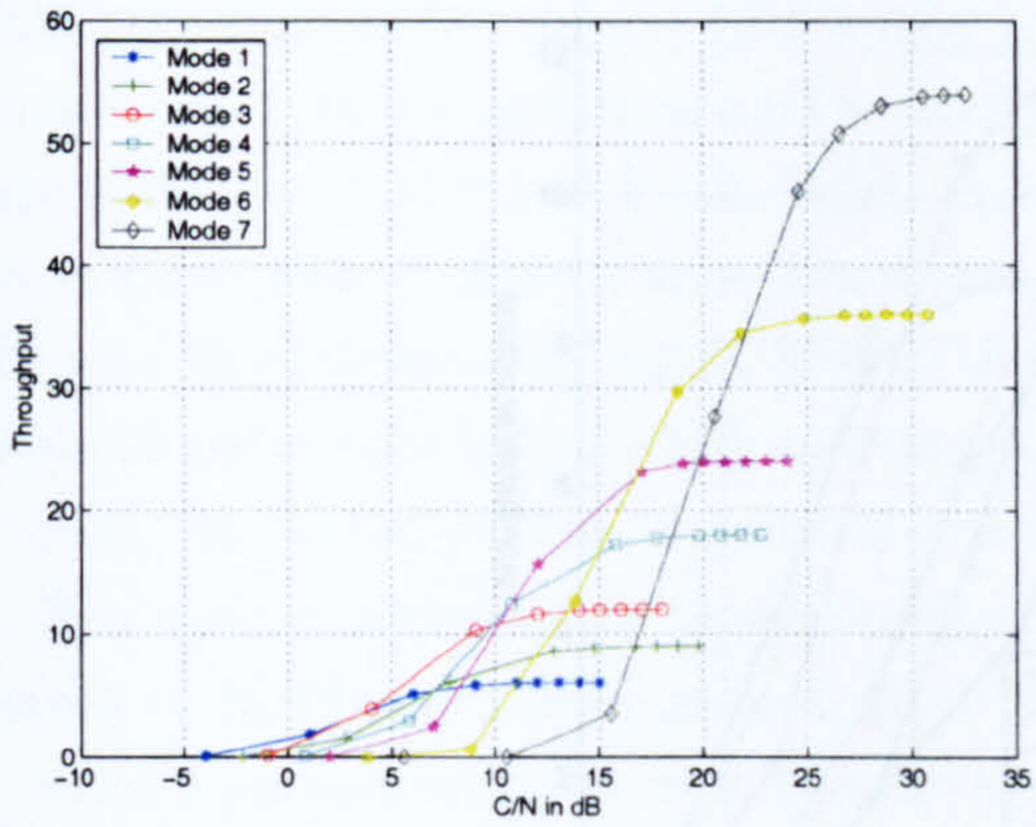
(d) PDF Delay- 2Tx-2Rx, RS(188,252,t=32)

Figure 5.14: ARQ and delay Performance of the system for mode 3 - C/N = 2.5dB

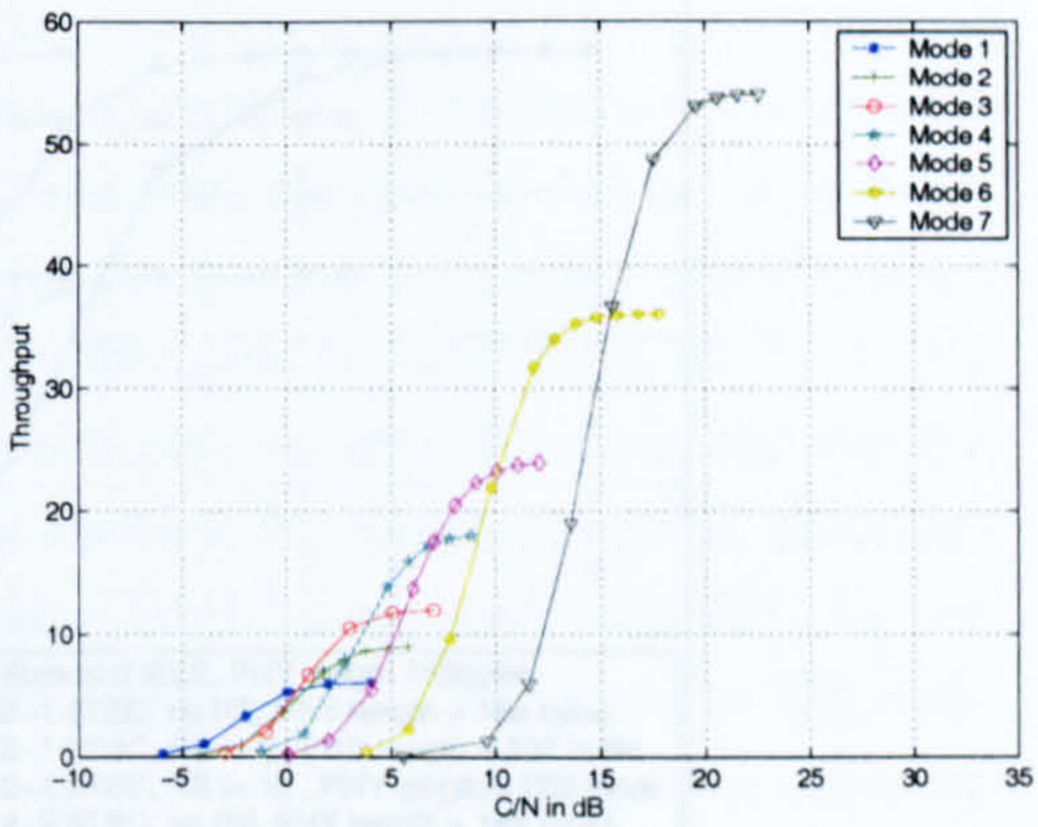
5.4.4 Throughput Performance

As mentioned in table 5.1, the 8 and 32 byte correction capabilities of the concatenated encoders lead to coding rates of 0.92 and 0.74 respectively. This will reduced the PHY layer nominal bit rate. Figure 5.15 shows the throughput performances for all the modes and for the 2Tx-2Rx configuration. The maximum link throughput, corresponding to an error free environment, is lower for RS(188, 252, t=32) than for RS(188, 204, t=8) since more PHY data is sent for the same amount of application data. For example, the nominal bit rate for mode 3 is 12 Mbits/s, and the maximum throughput achievable with this proposed system is around 9 Mbits/s for a correction capability of 32 bytes, whereas the 8 byte correcting code allows a maximum of 11 Mbits/s [138].

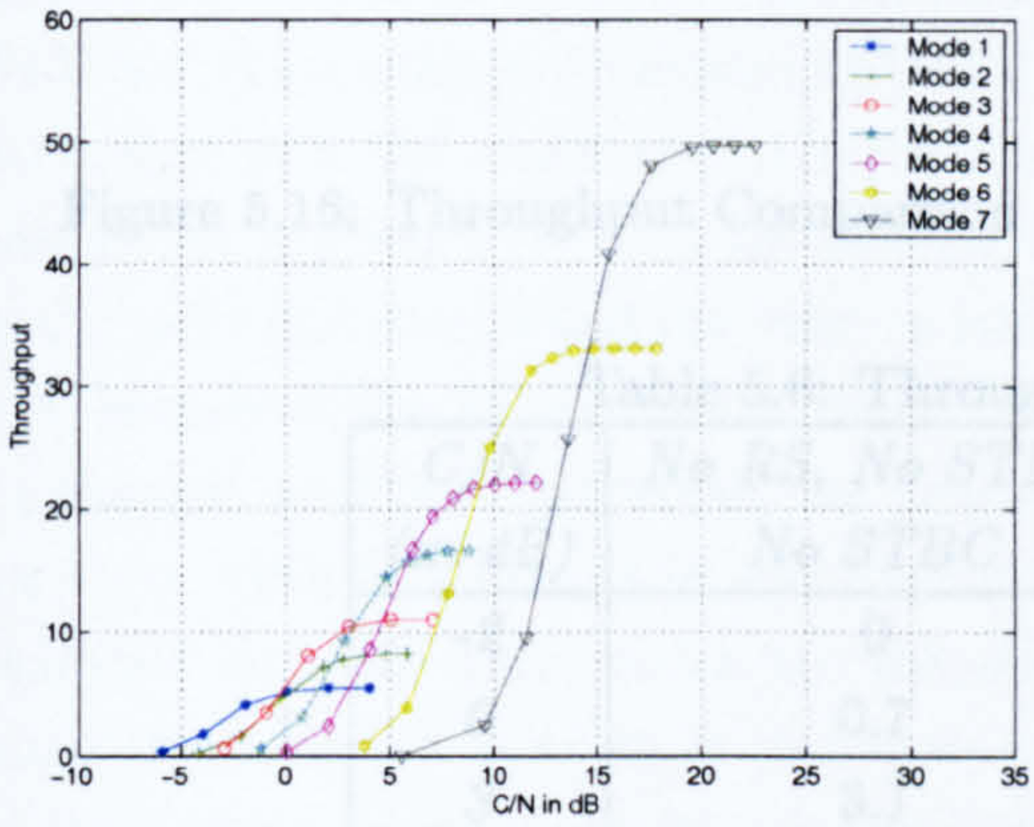
However, when the channel conditions are not good anymore, errors start occurring, influencing the throughput. Figure 5.16 shows the different throughputs achievable with 2Tx-1Rx and 2Tx-2Rx and with the two correction capabilities for mode 3. This is also summarised in table 5.6. It can be seen that for a low C/N value, i.e. bad channel conditions, the proposed combined system provides a better throughput. For a given



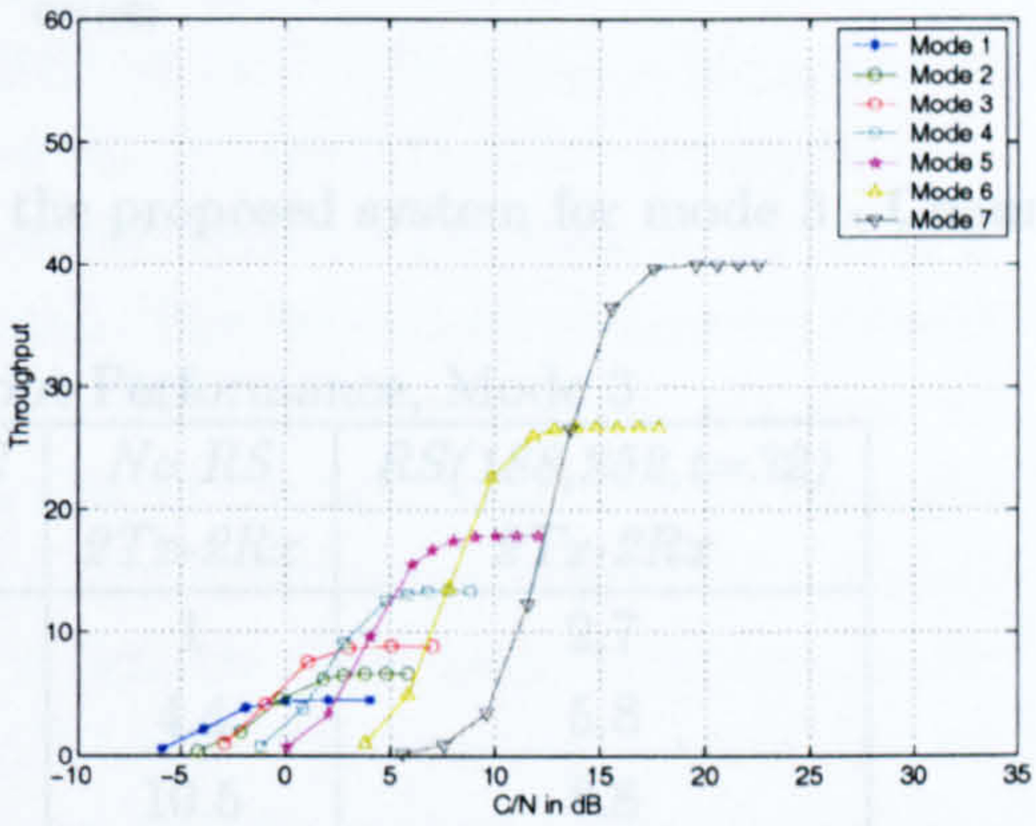
(a) no RS, no STBC



(b) no RS, 2Tx-2Rx



(c) RS(188,204,t=8), 2Tx-2Rx



(d) RS(188,252,t=32), 2Tx-2Rx

Figure 5.15: Link Adaptation for a 2Tx-2Rx STBC and different correction capabilities

STBC configuration, the RS code with the highest correction capability (32) provides the highest throughput for low C/N values. However, since this correction capability has a lower maximum throughput, smaller correction capabilities give better throughput as the C/N increases. This means that as the C/N increases, the PER decreases and a stronger RS code is not necessary any more. A lower correction capacity code would now provide better throughput performance. Finally, as the PER gets closer to zero, the use of RS codes are no longer needed to provide good throughput performance [138]. An adaptation algorithm would therefore switch on RS and STBC in order to provide the highest throughput depending on the C/N.

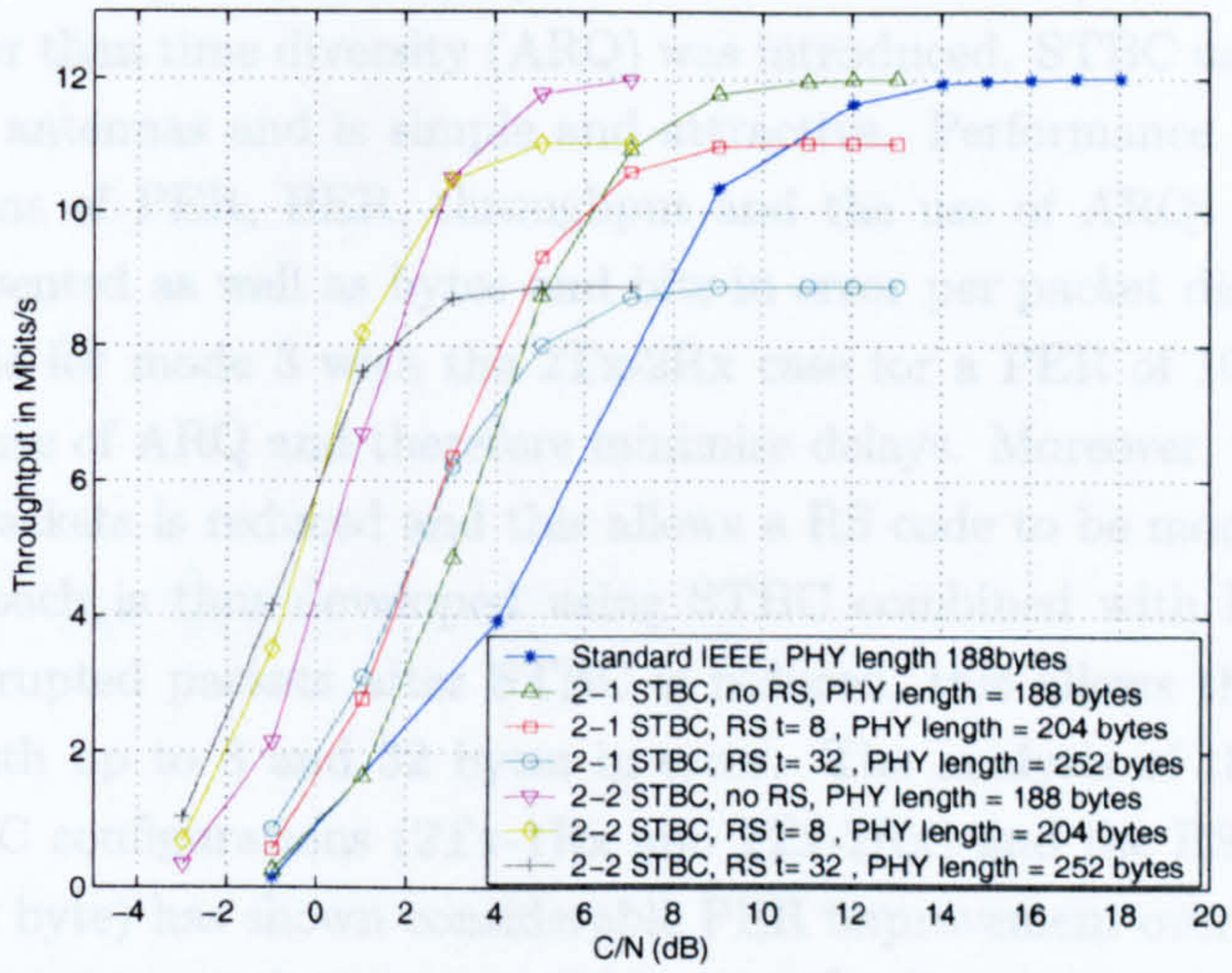


Figure 5.16: Throughput Comparison of the proposed system for mode 3 - Channel A

Table 5.6: Throughput Performance, Mode 3

C/N (in dB)	*No RS, No STBC* *No STBC*	*No RS* *2Tx-2Rx*	*RS(188,252,t=32)* *2Tx-2Rx*
-2	0	1	2.7
0	0.7	4.4	5.8
3	3.1	10.5	8.8
6	6.6	11.9	9
10	10.8	12	9

5.5 Conclusion

In this chapter, possible enhancements for MPEG-2 video transmission over the IEEE 802.11a/g PHY layer were studied in order to reduce the use of layer 2 ARQ. Because of delays, real-time MPEG-2 video transmission cannot tolerate significant MAC level retransmission. ARQ is however a mandatory feature of the IEEE 802.11 MAC layer which sits on the top of the IEEE 802.11a PHY layer. To guarantee MPEG-2 streaming over COFDM based WLANs without a strong reliance on the MAC ARQ, the PER at the PHY layer must be reduced.

A first approach using concatenated Reed Solomon codes (RS(188,204,t = 8) and RS(188,252,t = 32) with the convolutional code of the standard IEEE 802.11a PHY has been studied. This solution is very similar to the one successfully developed for the DVB-T and DVB-S standards, both using a COFDM based PHY layer. However, in an office environment, that does not provide Line of Sight and where WLANs are deployed, the conducted study showed that packets can get highly corrupted and cannot be corrected by the use of FEC. The PER is only slightly improved and the BER shows no improvement. The use of RS on its own does not provide a suitable solution to significantly reduced the number of ARQs.

To reduce heavily corrupted packets, Space Time Block Code (STBC) providing spatial diversity rather than time diversity (ARQ) was introduced. STBC uses multiple transmit and receive antennas and is simple and attractive. Performance results and improvements in terms of PER, BER, throughput and the use of ARQs along with delays have been presented as well as bytes and bits in error per packet distributions. 8dB gain is achievable for mode 3 with the 2Tx-2Rx case for a PER of 10^{-2}. STBC allows to reduce the use of ARQ and therefore minimise delays. Moreover, the density of highly corrupted packets is reduced and this allows a RS code to be more effective.

The second approach is thus developed using STBC combined with RS. As the density of highly corrupted packets after STBC is reduced, this allows the RS code to correct packets with up to 8 and 32 bytes in error. The analysis of the different combinations of STBC configurations (2Tx-1Rx and 2Tx-2Rx) and the RS correction capabilities (8 and 32 byte) has shown considerable PER improvement over the simple IEEE 802.11a/g standard (around 10dB for a PER of 10^{-2}). The number of packets to retransmit with the mandatory ARQ feature of the IEEE802.11 MAC is considerably reduced for a given C/N with the studied combined system. The reliance on ARQ is thus greatly diminished and the delay is minimised. The throughput evaluation showed that, because of the RS overheads, the error free transmission rate is lower. However, as the channel conditions get worse, the proposed scheme strongly improved the throughput since ARQ is reduced and less bandwidth is consumed for retransmission. STBC combined with RS outer codes are therefore a possible PHY layer enhancement for future multimedia transmission standards over WLANs. The proposed system enables reliable video transmission with a minimised use of ARQ by improving the performance of the standard IEEE802.11a/g.

Chapter 6

Video Packetisation

This chapter investigates packetisation issues and provides a cross-layer analysis for robust video transmission with emphasis given to broadcast transmission to several handheld devices. The chapter is organised as follows. Section 6.1 introduces the chapter. Related work is presented in section 6.2. Section 6.3 gives a brief overview of the system and the protocol stack. The possible scenarios and simulation setups are defined in sections 6.4 and 6.5 respectively. Section 6.6 details the proposed packetisation strategies and the framework for the transmission of video sequences, including the proposed modifications to the current IEEE 802.11 MAC specifications, as well as simulation results. A cross-packet FEC strategy is detailed in section 6.7. Finally section 6.8 concludes the chapter.

6.1 Introduction

Home entertainment, video on demand and other home multimedia communication products are now receiving considerable interest. New generations of handheld devices, such as personal digital assistants (PDAs), are now available to support good video resolution, and they provide connectivity that could be used for live video-like telesurveillance or live broadcast events. The IEEE 802.11 *Medium Access Control* layer (MAC) [39] is employed by all 802.11a/b/g products. This MAC has been studied in detail in chapter 3. It uses the *Distributed Coordination Function* (DCF), which is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), to gain access to the medium. Because of the high bit rates provided at the PHY layer (up to 54 Mbits/s), video over wireless LANs is about to become a reality. However, video is very sensitive to errors, and many problems remain, such as PHY performance and adaptation, variable end to end delay and jitter, and poor network Quality of Service (QoS) that may degradate the received video quality. Moreover, as already developed in 3, the IEEE 802.11 MAC has poor throughput efficiency, as shown in figure 6.1, and makes poor use of the high rate offered at the PHY layer. The throughput depends on the length of the MAC frame [80]. Small frames have a bad channel utilisation and, due to large overheads, they provide a very low throughput. Larger frames offer a better channel utilisation, and are therefore preferable for high bit rates application such as

video.

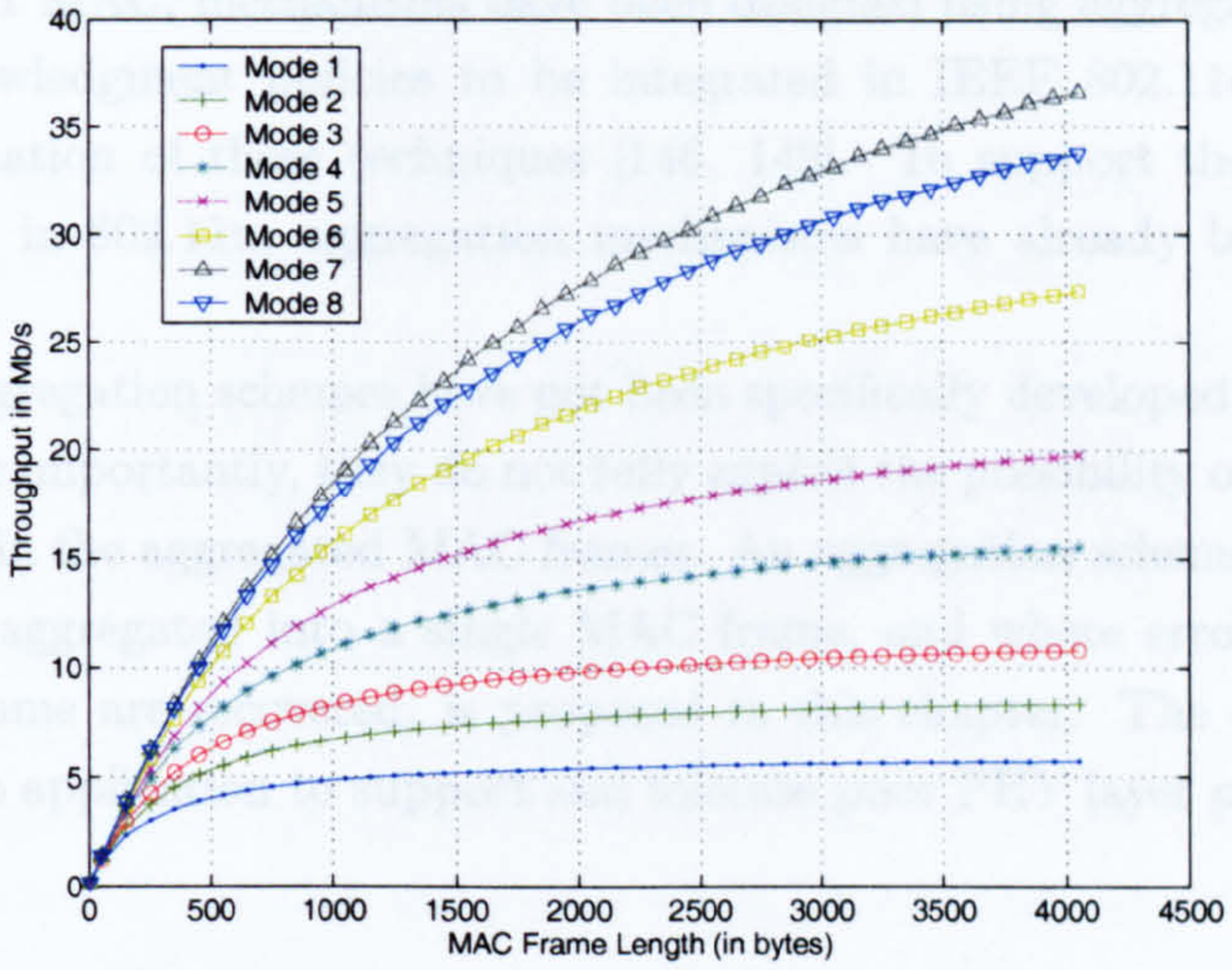


Figure 6.1: MAC Frame length Dependent Throughput of the IEEE 802.11 MAC

Because of its lack of QoS support[85, 146], the IEEE 802.11 task group is working on improvements under the IEEE 802.11e standard [40]. These include the enhanced DCF (EDCF), the use of frames transmitted in bursts and acknowledged using one single ACK frame (*Block_Ack*), as developed in [23], or the use of access categories (AC) with multiple queuing in order to differentiate services. The IEEE 802.11 task group is also currently in the process of proposing new MAC and PHY solutions in order to support higher bit rates (up to 216 Mbits/s) for the IEEE 802.11n standard [147]. The IEEE 802.11n MAC is expected to follow the basis of the IEEE 802.11e MAC with a *Block_Ack*-like scheme, but with the major addition of an aggregation mechanism. The final IEEE 802.11n standard is planned to be specified by the end of 2006.

The new H.264/Advanced Video Coding (AVC) standard [36] has been detailed in chapter 4. The output of the H.264/AVC is a *Network Abstraction Layer* (NAL) unit, containing a video slice. As previously explained, the slice structure is used for error resilience purposes. The concealment, or recovery, of a lost NAL unit (slice) is made easier if the NAL unit is small. Moreover, a smaller NAL unit contains less information and its loss is therefore less damaging, however this results in a higher slice overhead [77]. Moreover, slices provide points of temporal and spatial resynchronisation, where the decoding can resume whenever the received video stream is corrupted. For video robustness purposes, it is preferable to have small sized NAL units [12]. However, small NAL units encapsulate into small MAC frames, and, for MAC efficiency reasons, this is not desirable.

Many papers in the literature focus on methods to improve QoS, such as traffic prioritisation or scheduled access for applications competing for a shared resource. These methods are also being developed within IEEE 802.11e and are supported by the Wi-Fi Alliance and the Wi-Fi Multimedia (WMM) certification process [62]. The

study conducted in this chapter aims to improve real-time video transmission such that it can operate even with imperfect QoS. To overcome the poor throughput of the IEEE 802.11 MAC, mechanisms have been designed using aggregation [148], using modified acknowledgment policies to be integrated in IEEE 802.11e [23, 38], or by using a combination of these techniques [146, 149]. To support the very high data rates envisaged in 802.11n, aggregation mechanisms have already been proposed by TGnSync [147].

Previous aggregation schemes have not been specifically developed and deployed for video, and more importantly, they do not fully exploit the possibility of recovering error free IP packets in the aggregated MAC frames. An aggregation scheme, where multiple IP packets are aggregated into a single MAC frame, and where error-free IP packets in the MAC frame are recovered, is proposed in this chapter. The proposed scheme allows the video application to support and tolerate poor PHY layer performance. The proposal:

- guarantees good MAC throughput efficiency by ensuring good channel utilisation (through the use of large MAC frames).
- guarantees good video quality by maintaining video robustness with the use of small NAL units, and by retrieving possible error-free IP packets packets from an aggregated MAC frame.

Moreover, in order to offer robust video transmission and to support dropped or missing packets in the receiver, a packetisation strategy is developed at the application layer using cross-packet Forward Error Correction [50] to reconstruct missing NAL units.

6.2 Related Work

This work is motivated by the poor throughput efficiency of the 802.11 MAC, especially for small MAC frames, which lead to large overheads. Poor throughput efficiency is a major problem for video transmission, especially at higher bit rates where bandwidth starvation and excessive delay can occur.

In [23] and [38], the authors desribe an optional *No_ACK* policy under development for the IEEE 802.11e MAC enhancement [40]. At the expense of greater degradations at high error rates, MAC frames are not acknowledged. This enhances the throughput performance of the system. In [150], the author tackles the problem of MAC efficiency by proposing a simple concatenation mechanism where multiple MAC frames are concatenated and transmitted as a single (but longer) frame. This mechanism reduces overhead and therefore improves the MAC throughput. However, no recovery system is proposed, and whenever a longer frame is lost due to collision or channel error, the multiple concatenated MAC frames are all lost. In [148], the author proposes to concatenate PHY encoded MAC frames (with their header and CRC) in a so called *PHYsuperframe* with only one PHY header and preamble in order to reduced the

overhead. The algorithm aims to achieve higher throughput by concatenating PHY packets and by transmitting the *super frame* when the link supports a high data rate transmission. Retrieval mechanisms for error free MAC frames within the super frame were not developed in [148].

In [146] and [149], the authors refer to MAC-level improvements to be added to a future version of the IEEE 802.11e MAC, where one IP packet is fragmented into several blocks, and then aggregated into a single MAC frame. Each of the IP fragments is FEC protected. The receiver keeps a copy of the correctly received IP fragments in the current frame after FEC decoding. The MAC frame is acknowledged by the receiver only if all the IP fragments of the MAC frame are reconstructed. If no acknowledgment is received, the transmitter reschedules the whole MAC frame and the receiver combines the stored blocks with error free blocks retrieved from the retransmitted frame. This mechanism therefore reduces the number of transmissions. However, this scheme only considered fragmented IP packets and there was no mechanism allowing for partial retransmission.

The concept of aggregated frames is a core element in the enhanced IEEE 802.11 Task Group n proposal for higher throughputs (proposed by TGn Sync) [147]. Two types of aggregation are proposed: the $A-MSDU$ where multiple IP packets can be aggregated into a single MAC frame, and the $A-MPDU$ where multiple MAC frames are aggregated into a single PHY packet. No mechanism is yet fully specified to retrieve error-free fragments, if any, in a corrupted MAC frame.

The above techniques overcome the poor efficiency of the IEEE 802.11 MAC by using aggregation/concatenation. However, they do not allow the recovery of error free segments, they do not limit the retransmission process to errorred segments (i.e. partial retransmission). Moreover, these schemes are not specially designed for video transmission. It should be noted that the RTP format for H.264 [130] offers an aggregation mechanism that allows larger RTP packets to be passed to the lower layers. This allows a higher throughput at the MAC layer. However, the MAC discards any corrupted MAC frames and the content of the aggregated RTP packets are lost at the RTP layer. It should also be noted that, while many papers in the literature use statistical error models for the IEEE 802.11 PHY, in this chapter a fully compliant 802.11a/g PHY layer simulator was used to recreate accurate bit and packet error patterns [44].

6.3 System Architecture

This section provides an overview of the components required in the protocol stack to transmit real-time video over Wireless LANs and provides the background material necessary to understand this proposal. H.264 video has been chosen with the packet-based RTP format

6.3.1 The Real-time Transport Protocol

RTP (Request for Comment 3550) [131] was originally defined to enable real-time multimedia applications to be sent over the Internet by providing a thin transport layer that different applications can build upon to cater for their specific needs [12]. The core document of the RTP standard [131] specifies those functions that are expected to be common amongst all applications where RTP is used. *RTP Profiles* define generals sets of variables and applications. Specific documents are then issued for all particular applications. RTP has been widely and commonly used in conjunction with the UDP/IP underlying protocols. The RTP format for H.264 [130] was designed so that the H.264 encoded output can be transmitted over packet-oriented networks and follows frameworks developed for H.261 [151], H.263 [152] and H.263+ [153] RTP specifications. In addition, the design of the H.264 Network Abstraction Layer has been performed with IP-based transmissions in mind [35].

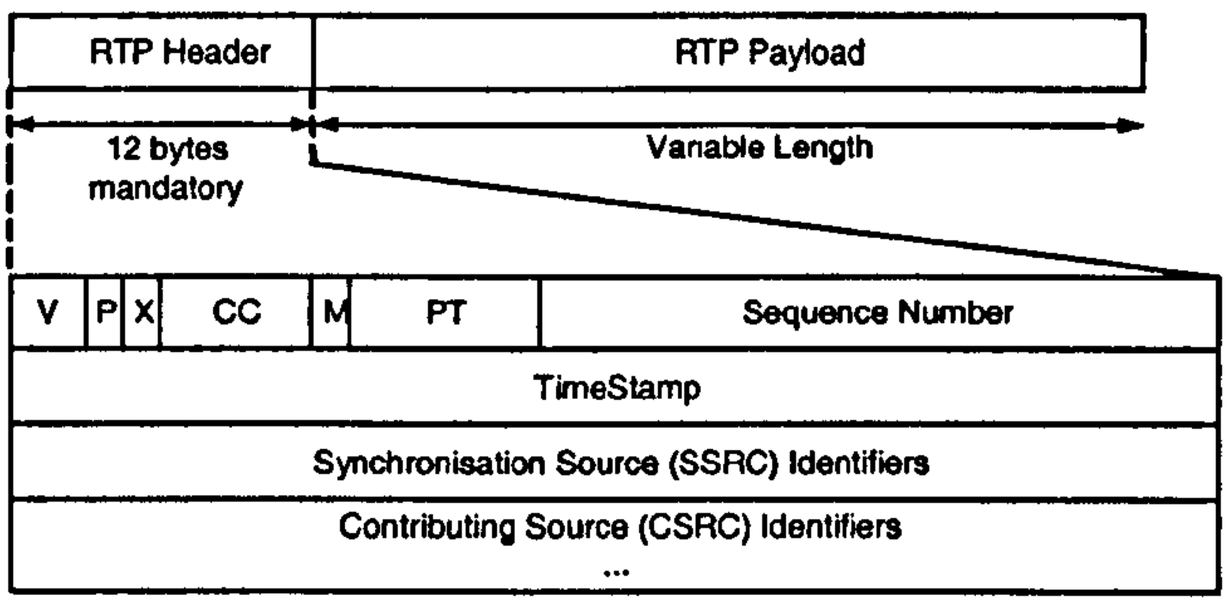


Figure 6.2: General format of a RTP packet

The general RTP packet format is described in figure 6.2. The payload has a variable length and no error correction or detection scheme is implemented. The RTP packet header comprises 12 fixed mandatory bytes and can be extended with optional fields. It is composed of:

- *V*: Version (2 bits) defines the version of RTP (2 is for the latest version, 1 is for the previous draft version, and 0 is for the original version).
- *P*: Padding (1 bit). If set to 1, the packet contains padding bytes at the tail that are not part of the payload.
- *X*: Extension (1 bit). If set to 1, the header is followed by exactly one header extension.
- *CC*: Contributing SouRCe (CSRC) Extension (4 bits) contains the number of CSRC identifiers following the fixed header.
- *M*: Marker (1 bit). Use is defined by a profile. It can be set to detect events such as frame boundaries.
- *PT*: Payload type (7 bits) identifies the format of the RTP structure. This may be set by profiles.

- *Sequence Number* (16 bits) is incremented by one for each RTP packet and may be used to detect packet loss and restore the original packet sequence.
- *Time Stamps* (32 bits) Reflect the sampling instant of the first bytes of the RTP data packet.
- *Synchronisation Sources ID* (32 bits) identifies the synchronisation sources.
- *Contributing Sources ID* (integer number of 32 bits: $n \times 32$, with n between 0 and 15.), identify the contributing sources for the payload in this packet.

RTP does not implement any form of transport level ARQ. The RTP payload specification for H.264 describes several packetisation schemes from NAL units to RTP packets [30, 130]. The basic mechanism is the mapping of one NAL unit to one RTP packet (*Single NAL unit* mode). The payload of the RTP packet is the NAL unit. The values of the RTP header are set as specified in RFC 3550 [131] and no provisions for the carriage of H.264 are necessary. With very small NAL units, such as parameters sets, the framework specifies an *Aggregation Packet* mode, where multiple NAL units can be aggregated into a single packet with four different versions [30]: Single-Time Aggregation Packet Type A (*STAP-A*), Single-Time Aggregation Packet Type B (*STAP-B*), Multi-Time Aggregation Packet with 16 bit offset (*MTAP16*) and Multi-Time Aggregation Packet with 24 bit offset (*MTAP24*). If its size is larger than the underlying maximum packet size allowed [30], the *Fragmentation Unit* mode allows a single NAL unit to be fragmented into multiple RTP packets with two different versions [30]: Fragmentation Unit Type A (*FU-A*) or Fragmentation Unit Type B (*FU-B*).

The H.264 RTP specification also defines three cases of packetisation mode: the single NAL unit mode, the non-interleaved mode and the interleaved mode. The non-interleaved and interleaved modes are allowed (or not allowed) depending on the packetisation schemes. The interleaved packetisation mode allows the transmission order of NAL units to differ from the decoding order by using the *Decoding Order Number* (DON), a field in the payload structure that indicates the NAL unit decoding order [130]. For the remainder of the chapter, the simple RTP packetisation case is applied, since with aggregation scheme combined with the IEEE 802.11 legacy MAC, the aggregated RTP packet would be missing if the MAC frame is missing.

6.3.2 Transport Layer: User Datagram Protocol

In an IP environment, the most common transport layer protocols are the Transport Control Protocol (TCP) [154] and the User Datagram Protocol (UDP) [64]. TCP offers connection-oriented and byte-oriented transmission that guarantees a reliable transport service. It is based on retransmission (ARQ) and time-out mechanisms [63][154]. However, ARQ retransmission is not suitable for all forms of time-bounded application, such as real-time video transmission, where low latency is essential.

UDP offers unreliable transport service without any retransmission mechanism (ARQ) at the transport layer. This provides a *best-effort* and connection-less ser-

vice. The general form of a UDP packet is shown in figure 6.3. The UDP header is 8 bytes long and the payload is of variable length. The first four bytes of the header are defined for the source and destination ports. The length of the UDP packet, including the header and the payload, is present in the header. UDP implements a check sum over the UDP header, the data and a pseudo header of information from the IP header, in order to detect erroneous packet. The pseudo header conceptually prefixed to the UDP header contains information such as source and destination addresses, the protocol and the UDP length. This information gives protection against misrouted datagrams [64]. This check sum is optional and can be turned off. A variant of UDP, called *UDP-Lite* implements a partial check sum that only covers as much of the user data that the sending application specifies as necessary [155, 156]

Only UDP will be considered in this chapter, in the context of a broadcast transmission, i.e. where no ARQ at any level can be used, and in the context of a low latency unicast transmission with no reliance on ARQ.

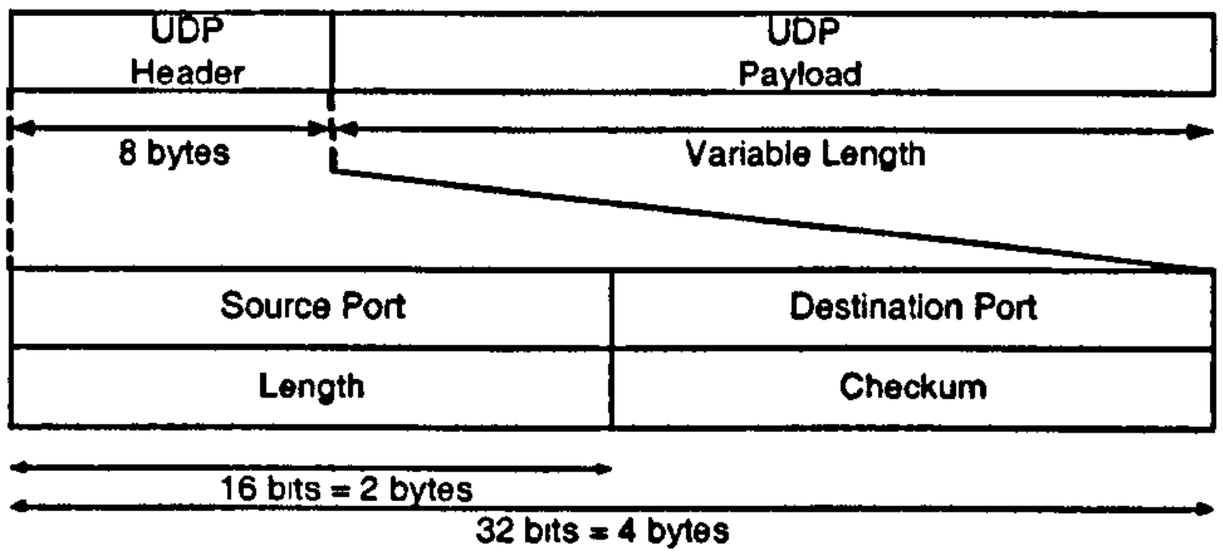


Figure 6.3: General format of a UDP packet

6.3.3 Network Layer: Internet Protocol

The Internet Protocol (IP) has been designed for use in interconnected systems of packet-switched computer communication networks [9]. This version of IP is also known as IPv4 (version 4). IP provides an unreliable, best-effort, connectionless packet delivery service and gives no guarantees on the actual delivery of the packet. The latest version of the protocol IPv6 (version 6) [157] enhances IPv4 by extending some addressing capabilities, simplifying the header format and by adding new security and privacy features. By IP, in this thesis IPv4 is assumed.

IP can perform fragmentation and packets can have variable length. If a packet coming from the transport layer is larger than the maximum permitted size at the IP layer (64kbytes) then it is fragmented. The fragmentation can however be tuned and is left to manufacturers. No ARQ is performed at the IP layer. The general IP packet format and its header are described in figure 6.4. In an IP packet, only the integrity of the header is guaranteed by a header check sum. No protection of the payload is performed. The mandatory header is 20 byte long. In this chapter, the RTP/UDP/IP packet is the payload of the MAC frame.

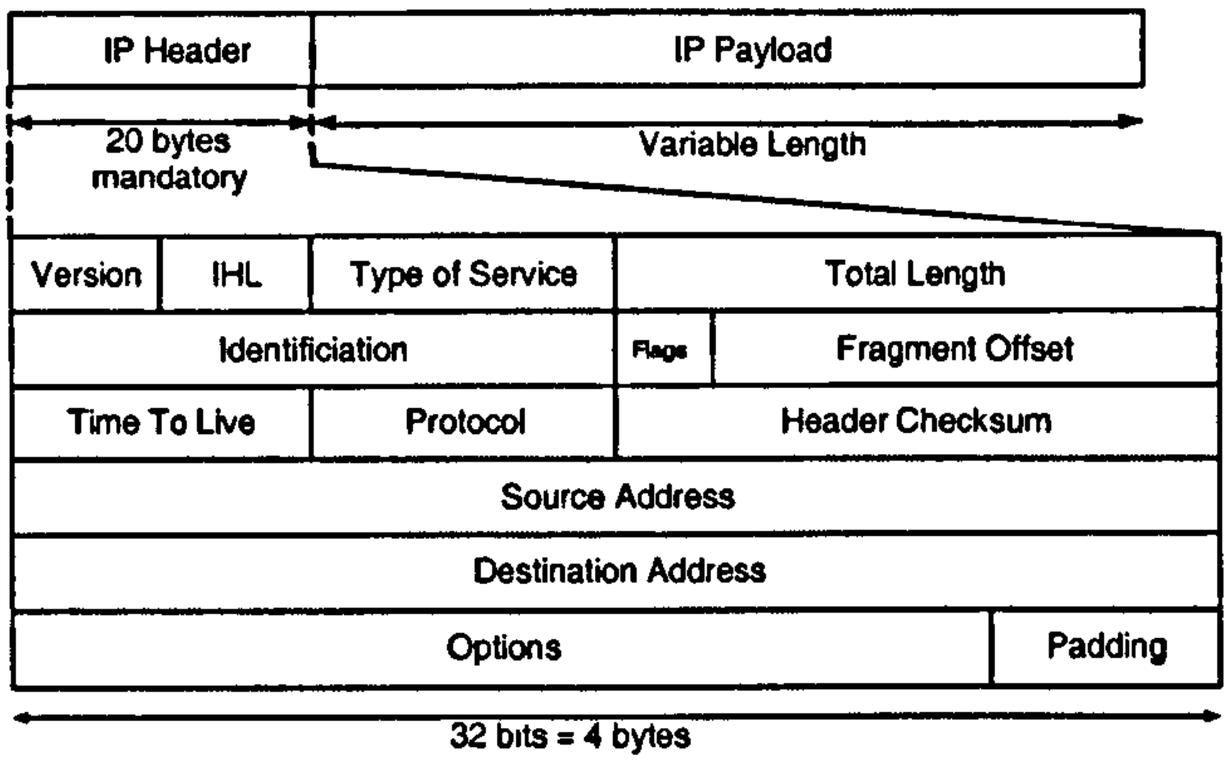


Figure 6.4: General format of an IP packet

6.3.4 IEEE 802.11 MAC and IEEE 802.11a/g PHY Layers

The IEEE 802.11 MAC and the IEEE 802.11a/g PHY layers have been deeply detailed in chapters 3 and 2 respectively. It is however recalled that the throughput at the MAC is dependent on the length of the MAC frame. The IEEE802.11 MAC relies on a *Stop and Wait* ARQ retransmission scheme. Successfully transmitted frames are acknowledged and if the receiver does not receive the ACK within a Short Inter Frame Spacing (*SIFS*), the frame is rescheduled. If after the maximum number of retransmissions allowed, the MAC frame is still not acknowledged, the frame is dropped. In this paper, we assume no collision, and the only scenario for which a frame needs to be rescheduled is that the received frame is erroneous. A 32 bit Frame Check Sum (FCS or CRC) calculated over the MAC header and the payload is appended at the tail of the body and is used to detect channel errors. The PHY BER performance is independent of the length of the PHY packet, whereas the PHY PER is PHY packet length dependent. A larger packet is more likely to be corrupted.

6.4 Scenarios

IEEE 802.11a/g has potential application in a number of scenarios, ranging from compressed HDTV redistribution in the home to high bit rate (and low delay) outdoor wireless cameras. More traditionally, this technology provides a data link for PDAs and laptops. In the latter case, bit rates per user are often low ($\leq 1 Mbits/s$) and a single channel is commonly shared between a large number of terminals; greatly increasing the likelihood of congestion and collision. For example, IEEE 802.11g provides only 3 channels, whereas IEEE 802.11a provides 12. Importantly, IEEE 802.11a/g is also likely to be used for Broadcast or Multicast applications, including live multimedia distribution or telesurveillance. In this case, no retransmission mechanism is allowed at either the transport or MAC layers. Moreover, the maximum number of ARQs allowed in the MAC is variable, with the exact number left to manufacturers. Values between 1 and 128 are permitted and 32 is commonly used as a default value by manufacturers.

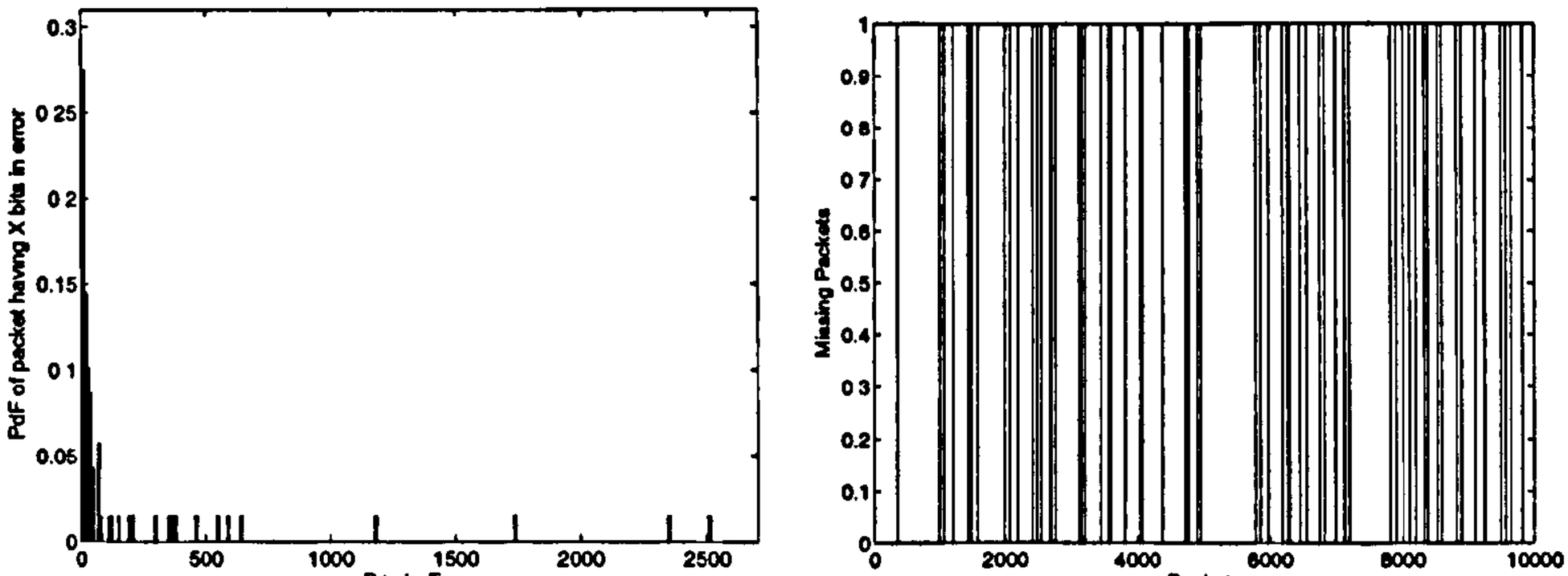
This value will influence the data throughput and delivery delay (see section 3.2.6.2 of chapter 3). To illustrate these scenarios, results are presented for RTP/UDP/IP links with no MAC layer ARQ (Broadcast/Multicast), or a limited number of MAC layer ARQ retries (Unicast).

6.5 Simulation Setups

While many papers in the literature use statistical error models of IEEE 802.11a/g or IEEE 802.11b simulators, results presented in this chapter have been obtained presented using an IEEE 802.11a/g PHY simulator. All results in sections 6.6 and 6.7 are obtained using error patterns generated by an IEEE 802.11a/g PHY simulator developed by Dr Angela Doufexi, Dr Michael Butler [44] and Dr Kamree Abul Aziz. The simulator supports all the operating modes and PHY layer packet lengths, and is capable of producing error patterns at any defined power level Carrier-to-Noise ratio (C/N). The channel model conforms to ETSI-BRAN channel A specifications (Non Line-of-Sight offce environment), with an *rms* delay spread of 50ns.

Figure 6.5(a) shows an example of the pdf distribution for the number of bits in error in the corrupted packets. Figure 6.5(b) shows PHY packet loss trace. Both results were generated using a PHY packet length of 256 bytes at a C/N of 11dB using mode 1 (PER=6.9×10^{-3}, BER=2.3×10^{-4}). Bit errors are clearly bursty within a packet, and it can be seen from the long distribution tail that a number of packets are heavily corrupted. In this case, the whole content of the PHY packet is lost. However, many PHY packets contain significant error free segments (especially those containing only a small number of errors). It should be noted that, in this study, uncorrelated channels have been used on a packet to packet basis at the PHY layer, meaning that each transmitted packet experiences a different channel. In practice, packets in error are correlated [158]. Because of the uncorrelated channel model, the packet loss trace shows a random distribution. A more realistic case with correlated channels was unfortunately not available. If it shows a random packet loss distribution, the uncorrelated channel however provides an accurate bits in error distribution within one packet. Analysis and integration of packet loss traces generated from real data measurements made using IEEE 802.11a/g cards has been carried out in chapter 8.

For the simulations, the *foreman* sequence has been encoded with 300 frames, with the sequence *IPPP...*, at CIF resolution (352×288 pixels) and at a frame rate of 30Hz using a modified version of the H.264 reference software version 7.3 [132]. At the encoder, the RTP format and a maximum fixed NAL unit size have been chosen. Generated slices are then encapsulated into UDP/IP packets. At the decoder, missing slices are concealed using the advanced error concealment algorithm of the reference software [16] as detailed in section 4.4.4. For a range of received C/N levels, the video sequence is sent 100 times in order to create statistical results. The PSNR values of the decoded sequences are then averaged. The 100 videos sequences were generated using different initialisation points in the error patterns. It is assumed that the sequence and picture parameter sets, which contain critical information about the video, are



(a) Bits in Error Distribution for Corrupted Packets

(b) Packet Loss Trace (1 = Packet Lost)

Figure 6.5: Error Model at C/N = 11dB, packet length = 256 bytes, mode 1

transmitted error-free using a TCP connection. Without this information, the decoder can not decode the received video sequence. In this chapter, mode 1 (BPSK 1/2 rate) is used.

The results are also representative of simulations at higher operating modes, at higher rates, and with larger MAC retry limit. They demonstrate a need for an error-free packet retrieval mechanism as explained in section 6.1.

6.6 Packetisation: From Video Packets to MAC frame

In recent years, many papers have proposed various solutions to optimise independently the different layers highlighted in the previous section. However, it is highly desirable to improve a specific layer with full knowledge of how this will impact on the other layers [50]. This is especially true for video transmission, where the content has a strong influence on performance, on performance, and where errors (or missing units) do not necessarily have an equal impact on the decoded video quality.

6.6.1 Transmission with the IEEE 802.11 Legacy MAC

A simple data encapsulation scheme from the application layer to the MAC layer is shown in figure 6.6. The MAC frame is contending for transmission over the wireless channel. Table 6.1 provides a summary of the ARQ of the ARQ and Check Sum options in the protocol stack (including the case of TCP). The overheads from the NAL unit payload to the MAC frame with CRC are in the order 75 bytes, with 40 bytes accounting for the RTP/UDP/IP layers [30, 125]. The reception scenario of the legacy MAC with a UDP link is as follows:

- The PHY layer decodes the received PHY packet.
- The MAC checks the FCS field to detect errors.

- If the FCS is correct, the MAC sends back an ACK and the MAC frame is de-encapsulated and passed up through the IP, UDP, RTP, NAL and application layers.

- If the FCS is not correct, the MAC does not sent an ACK. After one SIFS interval, the transmitter checks the re-transmission status of the frame (layer 2 ARQ algorithm):

  - If the number of retransmissions is less than or equal to the maximum retry count, then the frame is rescheduled.

  - If the number of retransmissions is greater than the maximum retry count, then the frame is dropped and the transmitter proceeds to the next frame.

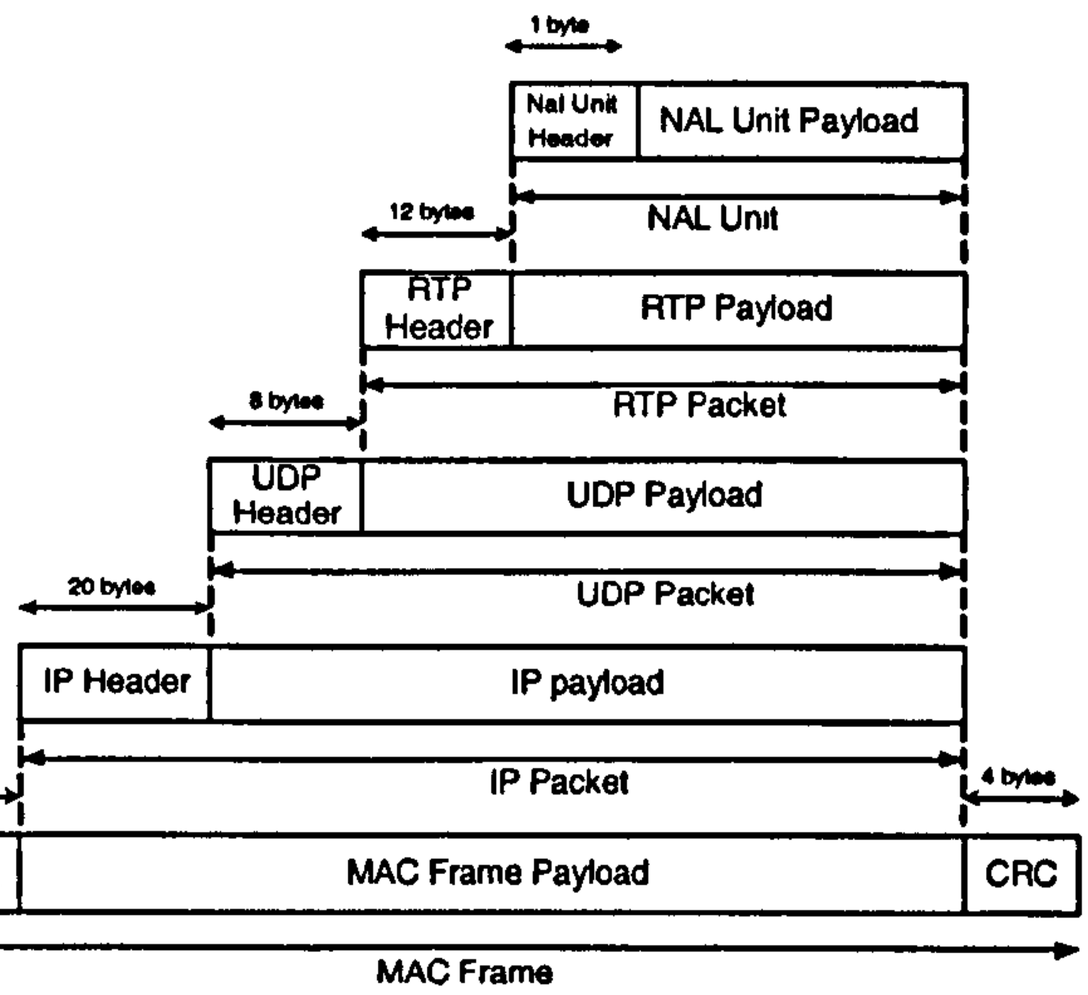


Figure 6.6: Simple Packetisation

Table 6.1: ARQ, Check sum and Header summary

Layer	*Packet*	*ARQ*	*Check Sum*	*Header (in bytes)*
Application	NAL Unit/RTP	No	No	1 (NAL)+ 12 (RTP)
Transport	UDP	No	Whole Packet	8
	TCP	Yes	Whole Packet	20
Network	IP	No	IP Header	20
Link/MAC	MAC Frame	Stop And Wait	Whole Frame	30 + 4 (CRC)

Using TCP, if the MAC frame is dropped, the TCP layer fails to receive an ACK for this packet. The TCP layer therefore reschedules the missing TCP packet, which once again will pass through the MAC and PHY layers.

6.6.2 Motivations

The proposal of packetisation presented in this chapter is based on the fact that a MAC frame can carry more than one NAL unit [76, 77]. To illustrate the benefits of multiple NAL units transmitted over a single MAC, as explained in the previous sections, the available throughput (or bit rate) available for different numbers of NAL units per MAC frame is shown in figure 6.7 (and table 6.2) for mode 1 of the IEEE 802.11a PHY layer with the IEEE 802.11 legacy MAC. When only one small NAL unit (125 bytes) is carried over a MAC frame, a very low throughput is observed. However, throughput increases as the number of NAL units carried over a single MAC frame increases (because of the larger MAC frame). For example, a 50% increase in throughput efficiency is achieved by mapping 4 NAL units (rather than a single NAL unit). The size of a MAC frame, and therefore the number of NAL units mapped into a MAC frame, is seen to be very important when evaluating the system capacity.

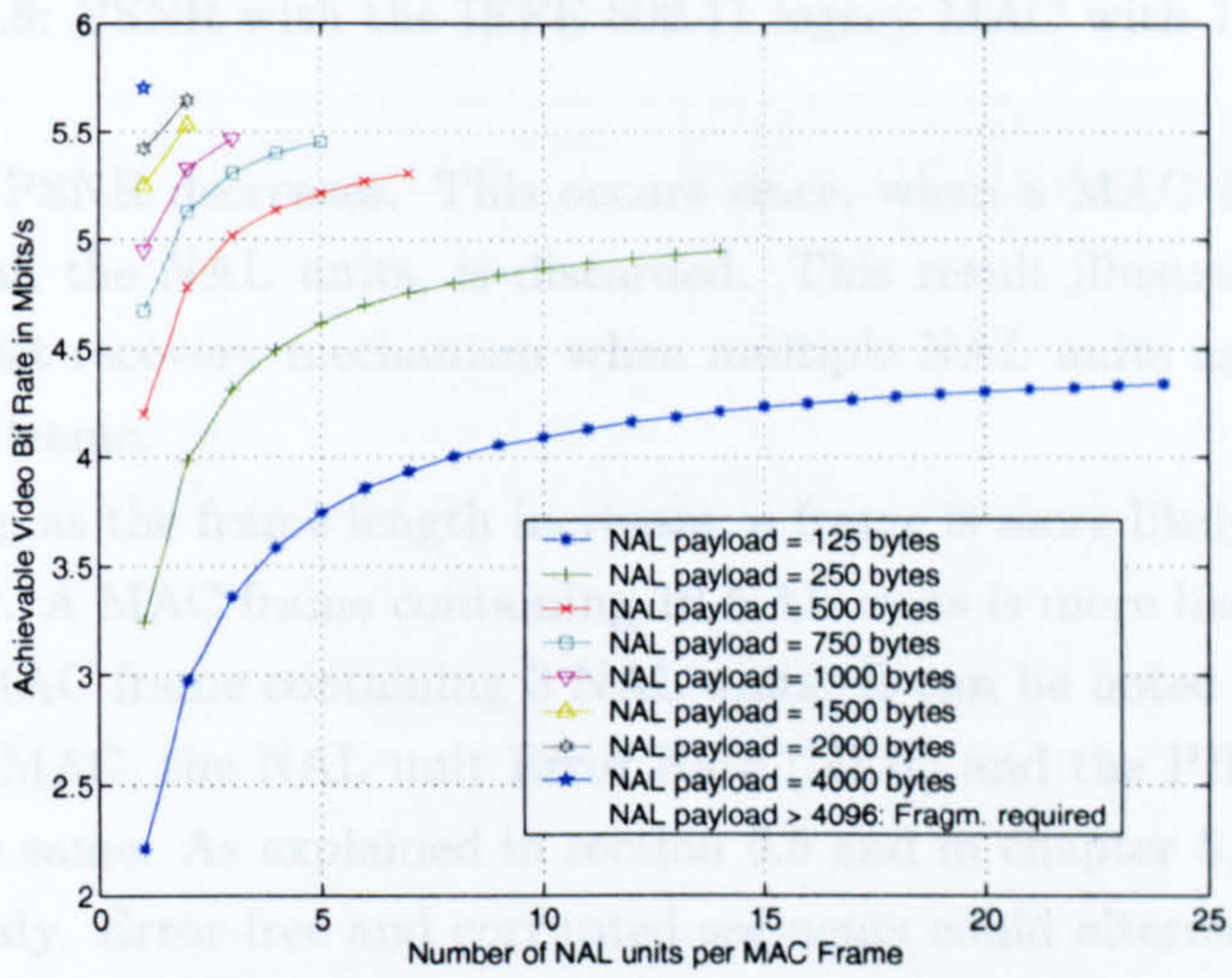


Figure 6.7: Throughput available for different number of NAL units per MAC Frame - Mode 1

Table 6.2: Maximum Throughput Comparison - Mode 1 - Fixed NAL Size - 125 bytes

Number of NAL units per MAC Frame	*Throughput in Mbits/s*
1	2.2
2	2.95
4	3.6
8	4

The way the NAL units are handled at the MAC layer determines the overall quality of transmission. More specifically, figure 6.8 depicts the effect on the *Peak Signal to Noise Ratio* (PSNR) of varying the number of NAL units aggregated into a single MAC frame (using the legacy MAC) for different received C/N. In this study, corrupted MAC frames, are discarded, after a single MAC ARQ. As the number of mapped NAL units

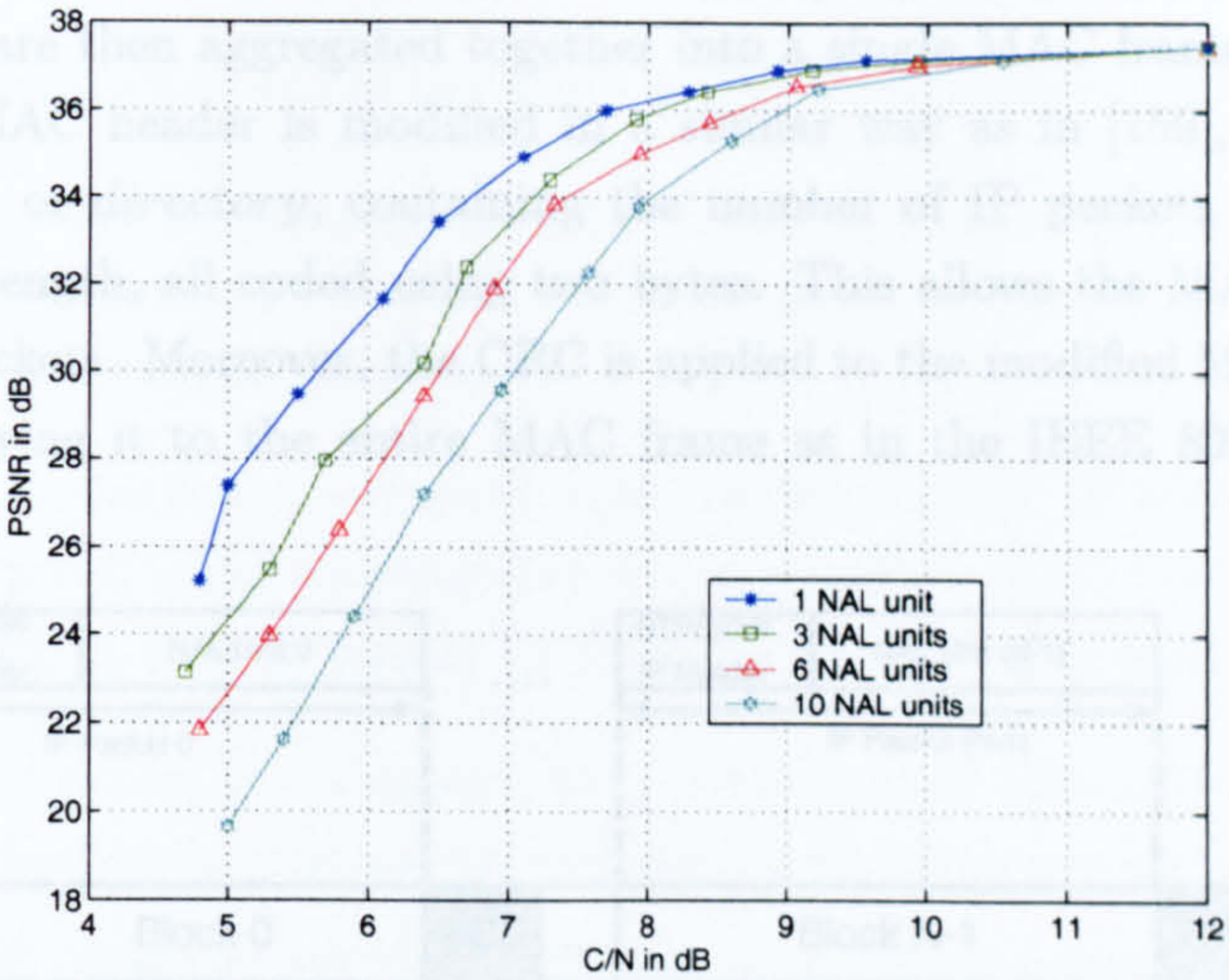


Figure 6.8: PSNR with the IEEE 802.11 legacy MAC with 1 ARQ allowed

increases, the PSNR decreases. This occurs since, when a MAC frame is lost, all the content, i.e. all the NAL units, is discarded. This result illustrates the need for an error-free packet recovery mechanism when multiple NAL units are transmitted using a single MAC frame.

In addition, as the frame length increases, a frame is more likely to be corrupted at the PHY layer. A MAC frame containing 10 NAL units is more likely to be received in error than a MAC frame containing 3 NAL units. It can be noted that with the IEEE 802.11 legacy MAC, the NAL unit Error Rate (NER) and the PHY packet error rate (PER) are the same. As explained in section 6.5 and in chapter 5, in wireless systems errors are bursty. Error-free and corrupted segments could alternate in a PHY packet depending on the channel conditions.

The main point of the proposal is to guarantee high throughput performance by mapping several small IP packets (NAL units) into a single larger MAC frame, as well as to guarantee video quality by recovering possible error-free IP packets in a corrupted MAC frame and maintaining the robustness of the video by using small NAL units.

6.6.3 Proposal for Packetisation/Mapping

A RTP/UDP/IP transmission is considered for a broadcast or unicast link. However, the presented mechanism remains valid for a TCP link. Mapping several NAL units (with RTP/UDP/IP encapsulation) allows higher throughput at the IEEE 802.11 MAC layer. Unfortunately, as mentioned in section 6.1, the legacy MAC drops the whole corrupted packet, even if the MAC frame contains error-free segments. The legacy MAC does not allow the retrieval of such error-free NAL units. Here, modifications to the IEEE 802.11 MAC layer that allow the recovery of error-free NAL units are proposed and the improvements in terms of video quality are assessed.

An IP packet carries a NAL unit with its RTP/UDP header. At the tail of each IP

packet a 4 byte long check sum field is appended as in [146] and [149]. These 'check summed' IP packets are then aggregated together into a single MAC frame, as shown in figure 6.9. The MAC header is modified in a similar way as in [159], by adding an *aggregationfield*, or *directory*, containing the number of IP packets aggregated and their respective length, all coded using two bytes. This allows the MAC layer to reconstruct the IP packets. Moreover, the CRC is applied to the modified MAC header only, instead of applying it to the entire MAC frame as in the IEEE 802.11 legacy MAC.

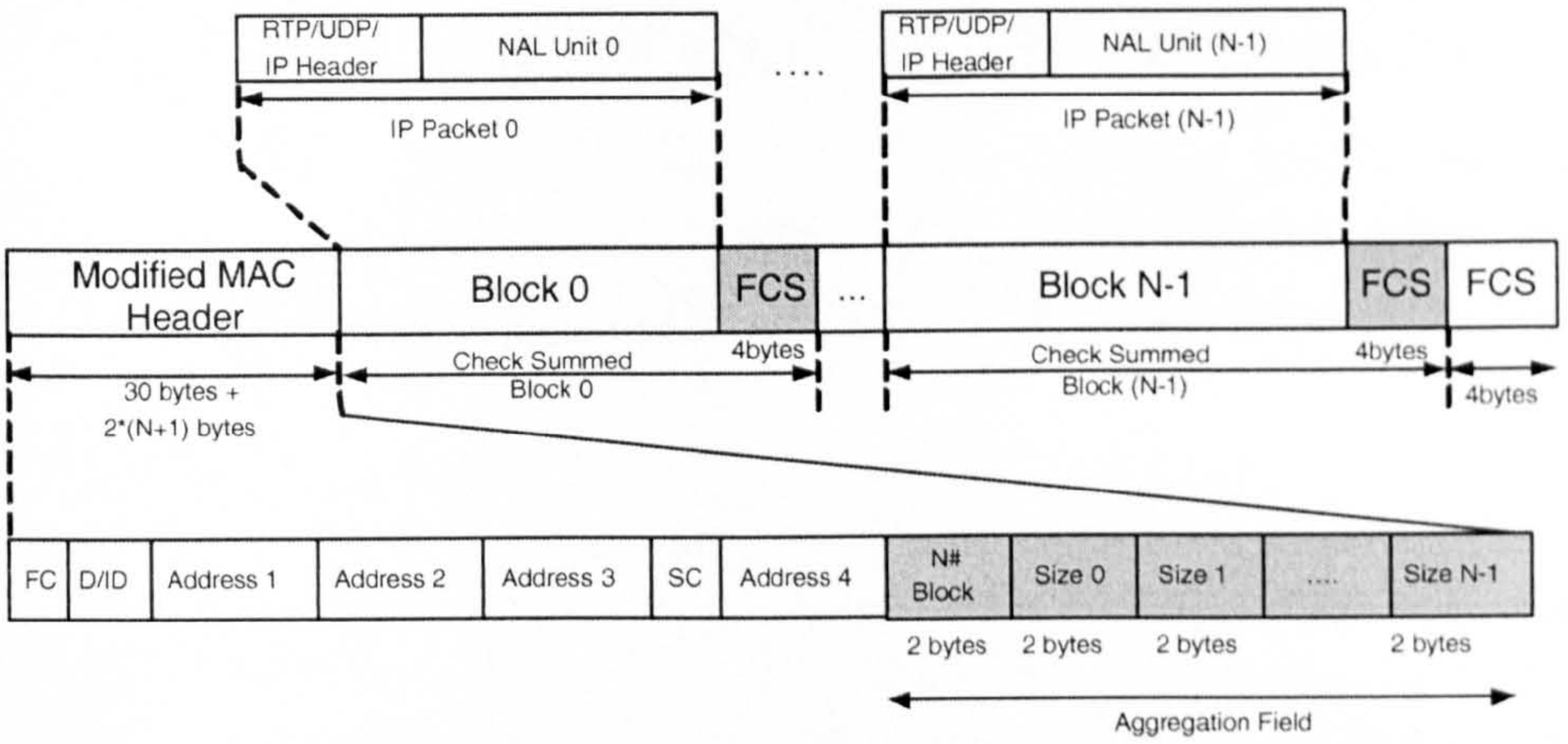


Figure 6.9: MAC Frame Modification for Multiple NAL Units MAC Packetisation

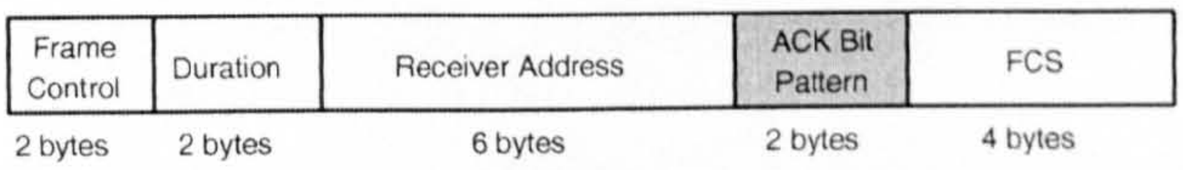


Figure 6.10: Modified ACK Frame

After reconstructing the IP packets and checking their FCS, the MAC at the receiver has knowledge of which IP packets require retransmission and which IP packets can be passed to the upper layers. This proposal therefore uses a selective repeat ARQ scheme within a MAC frame, which is similar to the *Block_ACK* of [23] and IEEE 802.11e [40]. The ACK frame should be modified so that it can incorporate a specific bit pattern (or *map*) describing the position of the IP packets to be retransmitted, again in a similar way to the *Block_ACK* policy. The modified ACK frame is shown in figure 6.10. A bit pattern is proposed with a '0' representing *'No Retransmission Required'* and a '1' representing *'Retransmission Required'*. The position of the bit corresponds to the position of the IP packets within the MAC frame (starting from 0). A 16-bit long pattern allows up to 16 IP packets to be mapped. For example, the following bit pattern 0001 1110 0000 0000 would be used to indicate that IP packets 3,4,5 and 6 require retransmission.

The scenario at the receiver is as follows:

- The PHY layer decodes the received PHY packet.
- The MAC checks the CRC of the MAC header to detect errors.
- If the MAC header is corrupted, then the frame is dropped (the MAC would not know if the destination and receive addresses are correction, or the start and the end of the aggregated IP packets).
- If the MAC header is not corrupted, each 'check summed' IP packet is identified and reconstructed using the directory in the modified MAC header. The MAC checks the FCS of each IP packet to determine whether it is corrupted or not:
  - If the IP packet is corrupted, then it is dropped (the IP layer would not know whether the destination and receive addresses are correct or not).
  - If the IP packet is not corrupted, the IP packet is passed through the IP, transport, RTP and application layers.
- The modified ACK frame is generated and transmitted.

The transmitter keeps a copy of all the IP packets mapped into each MAC frame in a buffer. The received ACK contains information on the position of the IP packets that require retransmission. The transmitter then updates its copy buffer by discarding correctly received IP packets, by keeping corrupted IP packets, and by including new IP packets. The transmitter keeps the retransmission IP packet status up to date, and discards IP packets when the maximum retry count is reached. In the case of a TCP link, the TCP layer ARQ process would be used to reschedule TCP packets that have not been acknowledged.

These modifications are not standard-compliant and add some complexity to the MAC. The overheads introduced by the bytes in the modified MAC header, the 2 bytes in the ACK frame, and the 4 byte FCS on each IP packet are insignificant compared to the frame length. The method can result in out-of-order packet delivery at the RTP layer as shown in figure 6.11. The sequence numbering mechanism available within RTP can bring NAL units back into decoding order. At the receiver, a jitter buffer is used at the application layer to deliver video packets to the decoder at the correct rate.

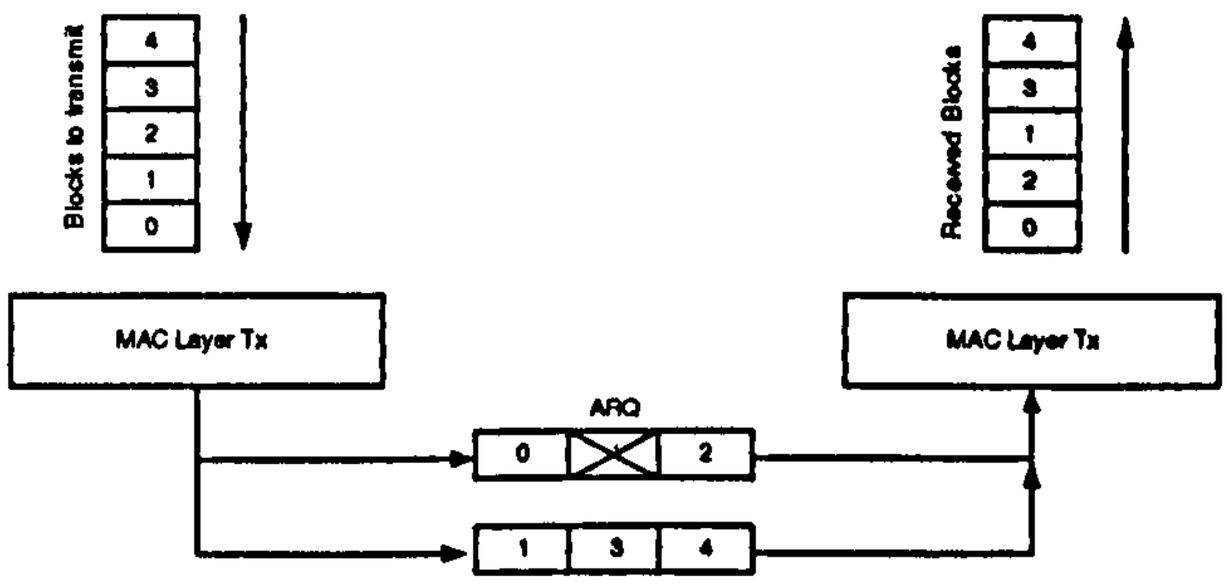


Figure 6.11: Out of Order Packet Reception

6.6.4 Improvements over the IEEE 802.11 Legacy MAC

This section compares the IEEE 802.11 legacy MAC, where packets are aggregated without recovery, with the proposed scheme in terms of NAL unit Error Rate (NER), throughput and PSNR and studies the influence of the number of fixed length NAL units (blocks) mapped into a single MAC frame. This simulation assumes that the video is pre-encoded and stored in a remote server. Encoding parameters such as the maximum NAL unit size (188 bytes) and bit rate (550kbits/s) are therefore fixed and can not be adapted in real time. Figures 6.12(a) and 6.12(b) show how one picture frame is affected when one MAC frame containing 4 NAL units is corrupted (the legacy MAC and for the proposed scheme respectively). With the legacy MAC, 4 NAL units are dropped, whereas the proposed scheme retrieves 2 error-free NAL units (and hence drops only 2 NAL units). The green part of the figure is due to the Y, U and V components being set to zero for missing data. Figures 6.12(c) and 6.12(d) show the 4dB of visual improvements on the received frames with the proposed system [76, 77] with a PER of 9.5×10^{-3} (after concealment is applied).

Figure 6.13 shows the achieved PSNR improvements for different C/N levels, over the legacy MAC, by using the modified MAC for the case of 3 and 10 NAL units aggregated respectively, and for the case of no ARQ (broadcast), and a maximum MAC layer retry limit of two (UDP Unicast). For a C/N of 4dB, the proposed scheme offers a 9dB gain in PSNR with 10 NAL units mapped and with two ARQs. Mapping 3 NAL units with two ARQs shows a 4dB gain for a C/N of 4dB in PSNR. It can be noted that when ARQ is used, re-transmission helps to reduce the PER for a given C/N and the PSNR curves are therefore shifted to the left.

Table 6.3 summarises the NAL Unit Error Rate (NER) and PSNR improvements achieved using the proposed scheme with 10 NAL units mapped into a MAC frame for Broadcast (i.e. no MAC layer retransmission). It can be seen that the proposed scheme improves the NER and the PSNR by up to 3dB for a PER of 10^{-2}, when 10 NAL units are mapped into one MAC frame [76].

Table 6.3: NER/PSNR Comparisons - 188 bytes per NAL units - 10 NAL units per MAC frame - Foreman - No ARQ (Broadcast)

		Legacy *MAC*		*Proposed* *Scheme*	
PER	*C/N*	*NER*	*PSNR*	*NER*	*PSNR*
10^{-2}	11.47dB	10^{-2}	30.79dB	3.9×10^{-3}	33.89dB
5×10^{-3}	13.03dB	5×10^{-3}	33.76dB	1.6×10^{-3}	35.82dB
10^{-3}	15.19dB	10^{-3}	36.56dB	1.4×10^{-4}	37.08dB

Figure 6.14 compares the NER and the PSNR of the proposal with the legacy MAC, for no ARQ (Broadcast/Multicast), for different C/N levels in dB. With the legacy case, for a given C/N level, the NER is equal to the PHY PER, and, because the MAC frame length increases as the number of NAL units mapped increases, the NER therefore goes up. On the other hand, the proposed scheme offers an almost constant NER as the number of NAL units mapped increases, and is equal to the PHY PER where only

(a) Legacy MAC

(b) Proposed MAC

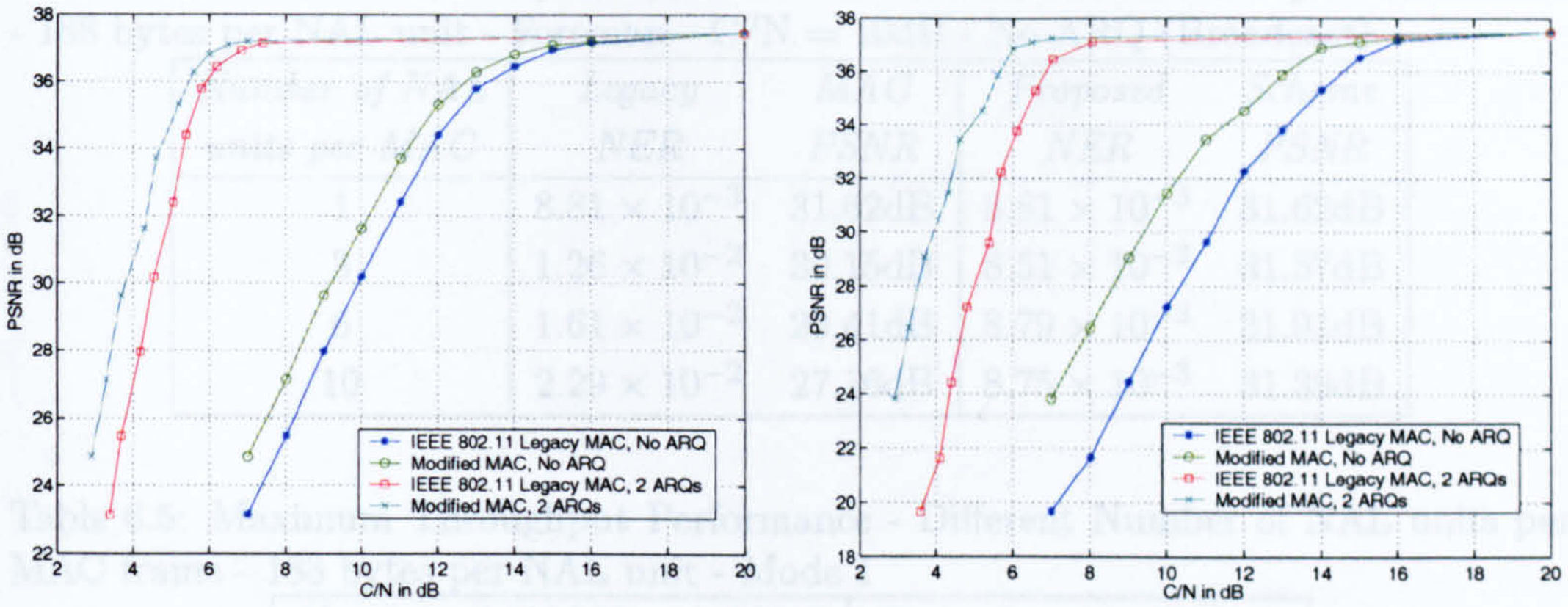
No Concealment

(c) Legacy MAC - PSNR = 27.50dB

(d) Proposed MAC - PSNR = 31.36dB

Concealment

Figure 6.12: 4 NAL units per MAC frame at PER = 9.5×10^{-3}

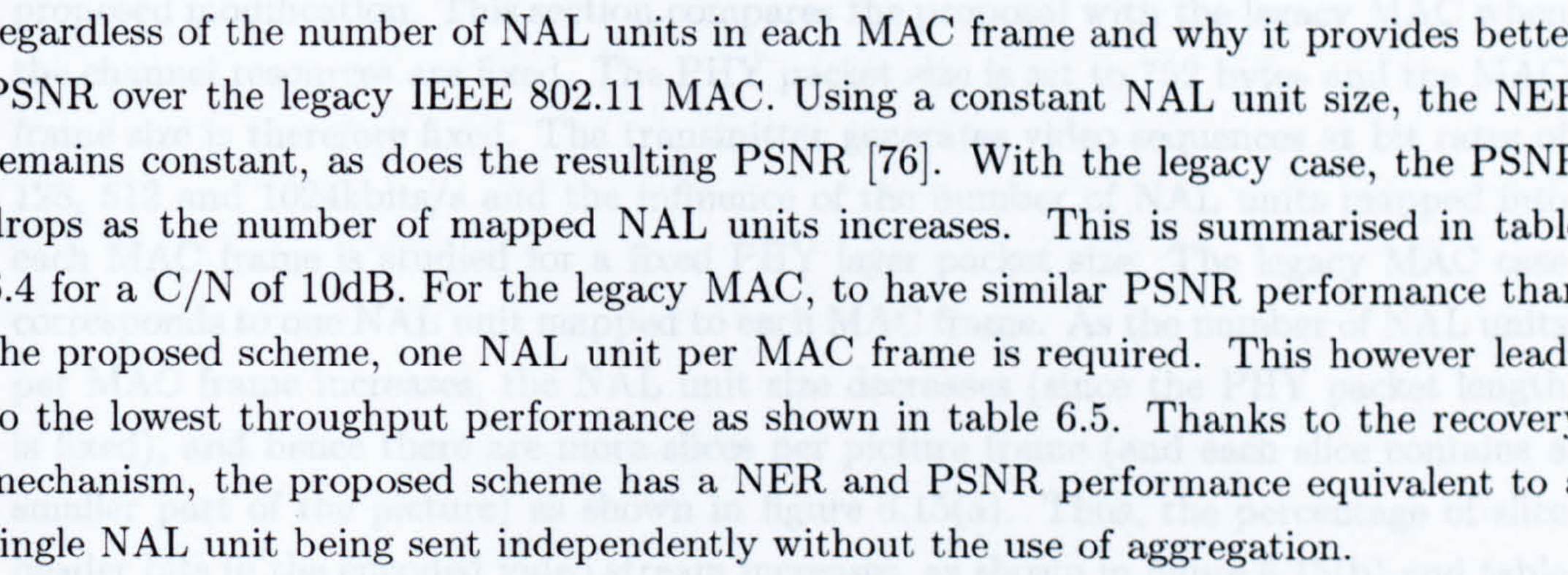


(a) 3 NAL units

(b) 10 NAL units

Figure 6.13: PSNR comparison of the IEEE 802.11 legacy MAC and the Proposed MAC, Fixed NAL unit sizes

one NAL unit is mapped. For a given C/N level, the PHY PER is MAC frame length dependent whereas the BER is not. With a constant NAL unit size, the NER remains therefore constant. This explains why the proposed scheme offers constant quality regardless of the number of NAL units in each MAC frame and why it provides better PSNR over the legacy IEEE 802.11 MAC. Using a constant NAL unit size, the NER remains constant, as does the resulting PSNR [76]. With the legacy case, the PSNR drops as the number of mapped NAL units increases. This is summarised in table 6.4 for a C/N of 10dB. For the legacy MAC, to have similar PSNR performance than the proposed scheme, one NAL unit per MAC frame is required. This however leads to the lowest throughput performance as shown in table 6.5. Thanks to the recovery mechanism, the proposed scheme has a NER and PSNR performance equivalent to a single NAL unit being sent independently without the use of aggregation.



(a) NER

(b) PSNR

Figure 6.14: Comparison of the IEEE 802.11 legacy MAC and the Proposed MAC, No ARQ (Broadcast), for different power levels - Fixed NAL Size

Table 6.4: NER - PSNR Comparisons - Different Number of NAL units per MAC frame - 188 bytes per NAL unit - Foreman - C/N = 10dB - No ARQ (Broadcast)

Number of NAL units per MAC	*Legacy NER*	*MAC PSNR*	*Proposed NER*	*Scheme PSNR*
1	8.81×10^{-3}	31.62dB	8.81×10^{-3}	31.62dB
3	1.26×10^{-2}	30.15dB	8.51×10^{-3}	31.57dB
6	1.61×10^{-2}	29.41dB	8.79×10^{-3}	31.91dB
10	2.29×10^{-2}	27.16dB	8.75×10^{-3}	31.39dB

Table 6.5: Maximum Throughput Performance - Different Number of NAL units per MAC frame - 188 bytes per NAL unit - Mode 1

Number of NALUs per MAC	*Throughput in Mbits/s*
1	2.79
3	3.94
6	4.37
10	4.57

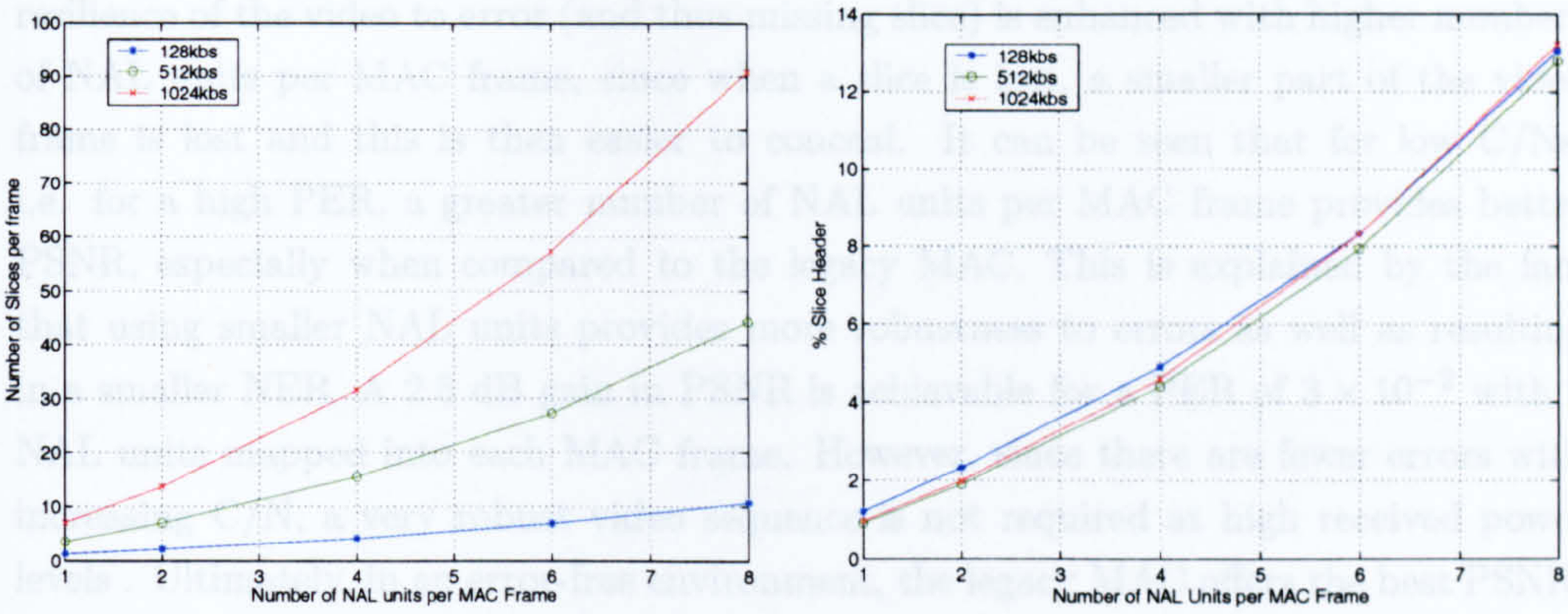
6.6.5 Robustness and Study with fixed MAC frame size

The previous section showed the improvements over the legacy MAC by using the proposed modification. This section compares the proposal with the legacy MAC when the channel resources are fixed. The PHY packet size is set to 752 bytes and the MAC frame size is therefore fixed. The transmitter generates video sequences at bit rates of 128, 512 and 1024kbits/s and the influence of the number of NAL units mapped into each MAC frame is studied for a fixed PHY layer packet size. The legacy MAC case corresponds to one NAL unit mapped to each MAC frame. As the number of NAL units per MAC frame increases, the NAL unit size decreases (since the PHY packet length is fixed), and hence there are more slices per picture frame (and each slice contains a smaller part of the picture) as shown in figure 6.15(a). Thus, the percentage of slice header bits in the encoded video stream increases, as shown in figure 6.15(b) and table 6.6 for the *foreman* sequence encoded at 128kbits/s. Increasing the number of NAL units per MAC frame reduces the NAL unit size and increases the robustness of the video at the expense of an increase in slice header. Similar figures are obtained with sequences encoded at 512 and 1024kbits/s.

Table 6.6: Slice statistics for *foreman* at 128kbits/s

Number of NAL Units per MAC	*Number of Slices Per Frame*	*Percentage Slice Overhead*
1	1.19	1.19
2	2.05	2.31
4	4.09	4.87
8	10.28	13.02

As the PHY packet size is fixed, the PHY PER does not depend on the number of



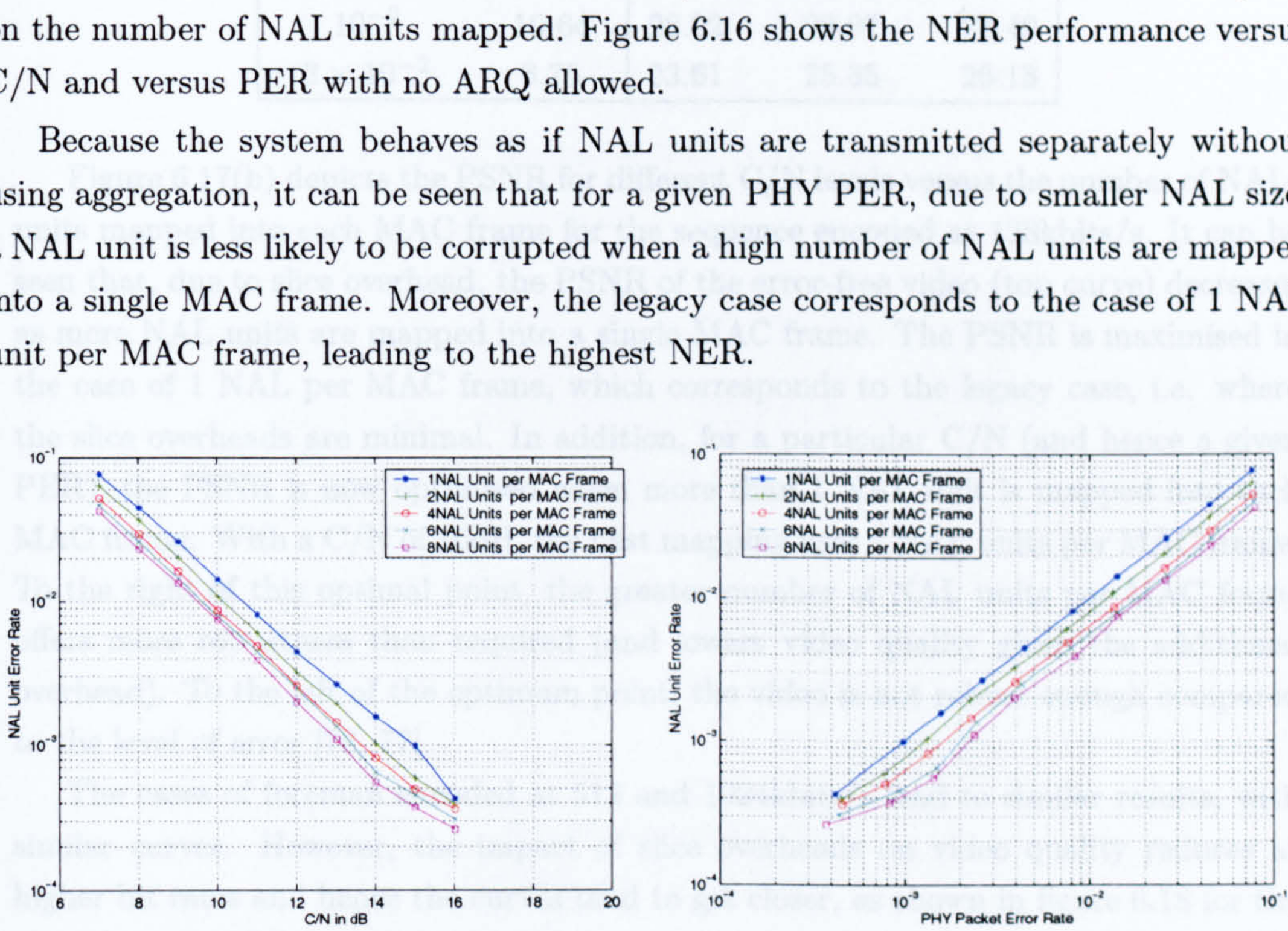
(a) Number of Slices per Frame

(b) Percentage of Slice Header

Figure 6.15: Influence of the number of NAL units per PHY packet, fixed PHY packet size = 752 bytes, Foreman

NAL units per MAC frame but only on the C/N level. The NER does however depend on the number of NAL units mapped. Figure 6.16 shows the NER performance versus C/N and versus PER with no ARQ allowed.

Because the system behaves as if NAL units are transmitted separately without using aggregation, it can be seen that for a given PHY PER, due to smaller NAL size, a NAL unit is less likely to be corrupted when a high number of NAL units are mapped into a single MAC frame. Moreover, the legacy case corresponds to the case of 1 NAL unit per MAC frame, leading to the highest NER.



(a) NER versus C/N

(b) NER versus PER

Figure 6.16: NER Performance for Foreman at 128kbs for the proposed MAC with no ARQ allowed - Fixed PHY Size

Figure 6.17(a) shows the PSNR for different C/N values for the *foreman* sequence encoded at 128kbits/s (transmitted with mode 1, see section 6.5) with the proposed MAC, and again with no use of ARQ. In the proposal, as the number of NAL units per MAC frame increases, the slice overhead increases (see table 6.6) and the error-free

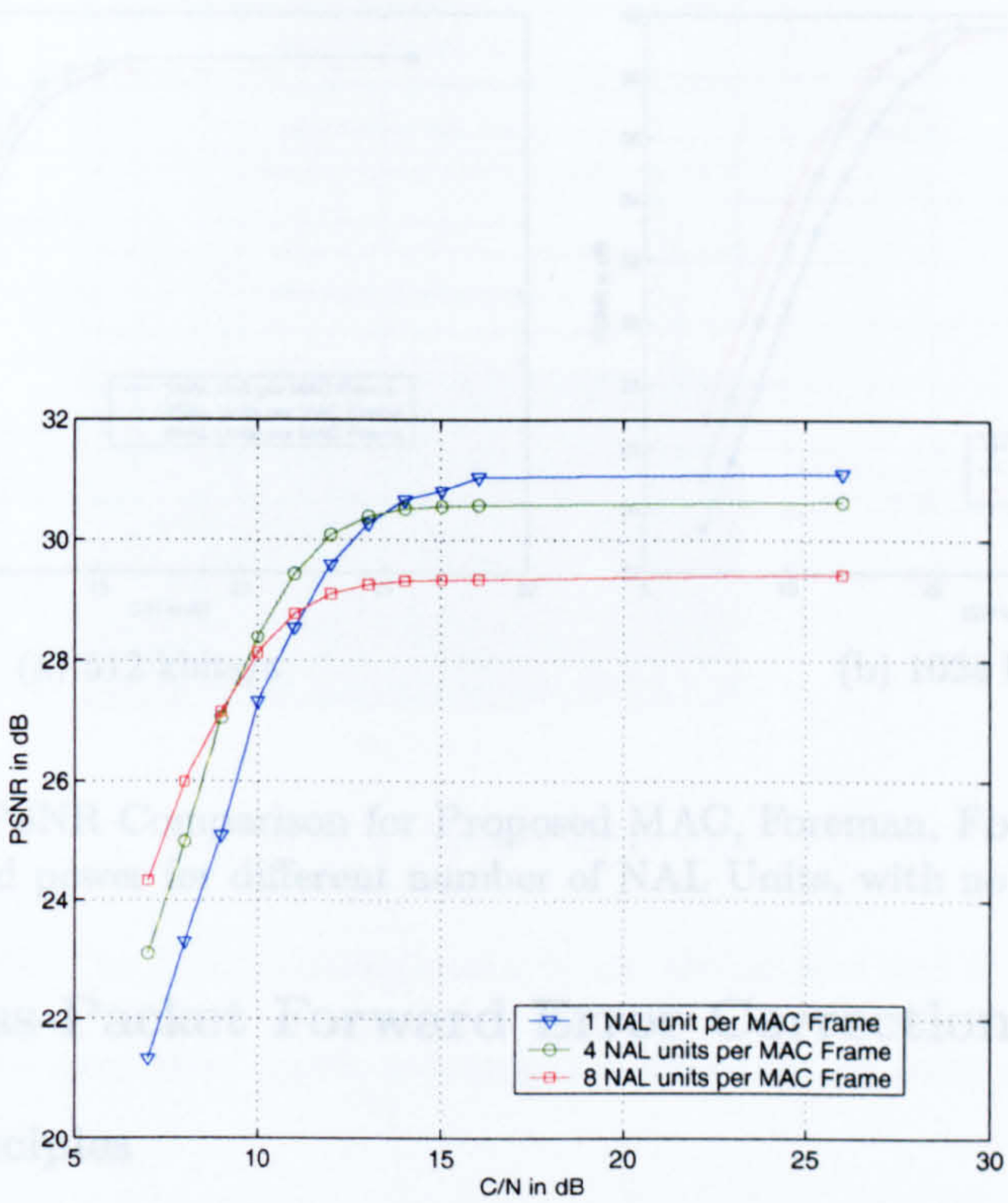
PSNR is therefore lower (since it is encoded here to a fixed video rate). However, the resilience of the video to error (and thus missing slice) is enhanced with higher numbers of NAL units per MAC frame, since when a slice is lost, a smaller part of the video frame is lost and this is then easier to conceal. It can be seen that for low C/Ns, i.e. for a high PER, a greater number of NAL units per MAC frame provides better PSNR, especially when compared to the legacy MAC. This is explained by the fact that using smaller NAL units provides more robustness to errors as well as resulting in a smaller NER. A 2.5 dB gain in PSNR is achievable for a PER of 3×10^{-2} with 8 NAL units mapped into each MAC frame. However, since there are fewer errors with increasing C/N, a very robust video sequence is not required at high received power levels . Ultimately, in an error-free environment, the legacy MAC offers the best PSNR. This is summarised in table 6.7.

Table 6.7: PSNR (in dB) Comparison - Fixed PHY Size

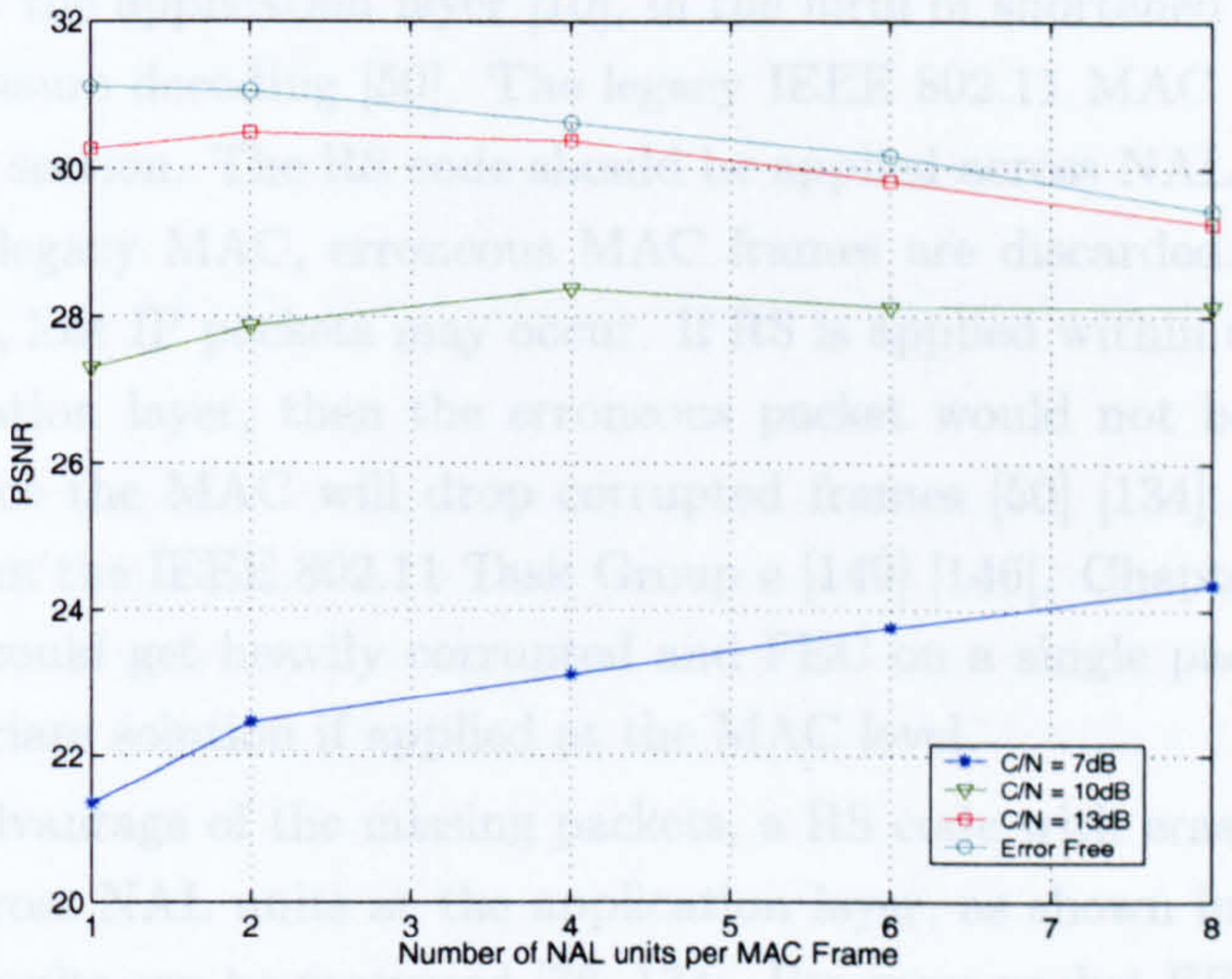
PER on the channel	*C/N (in dB)*	*NAL units*		
		1	*4*	*8*
0	-	31.09	30.61	29.42
10^{-3}	14.67	30.76	30.53	29.32
10^{-2}	10.64	28.02	28.97	28.49
3×10^{-2}	8.75	23.61	25.35	26.18

Figure 6.17(b) depicts the PSNR for different C/N levels versus the number of NALs units mapped into each MAC frame for the sequence encoded at 128kbits/s. It can be seen that, due to slice overhead, the PSNR of the error-free video (top curve) decreases as more NAL units are mapped into a single MAC frame. The PSNR is maximised in the case of 1 NAL per MAC frame, which corresponds to the legacy case, i.e. where the slice overheads are minimal. In addition, for a particular C/N (and hence a given PER), the PSNR is now optimised when more than 1 NAL unit is mapped into each MAC frame. With a C/N of 10dB, the best mapping uses 4 NAL units per MAC frame. To the right of this optimal point, the greater number of NAL units per MAC frame offers more robustness than required (and lowers video quality given the additional overhead). To the left of the optimum point, the video is not robust enough compared to the level of error [76, 77].

The cases of foreman encoded at 512 and 1024kbits/s lead to similar results, with similar curves. However, the impact of slice overheads on video quality reduces at higher bit rates and hence the curves tend to get closer, as shown in figure 6.18 for the video bit rate of 512 and 1024 kbits/s.

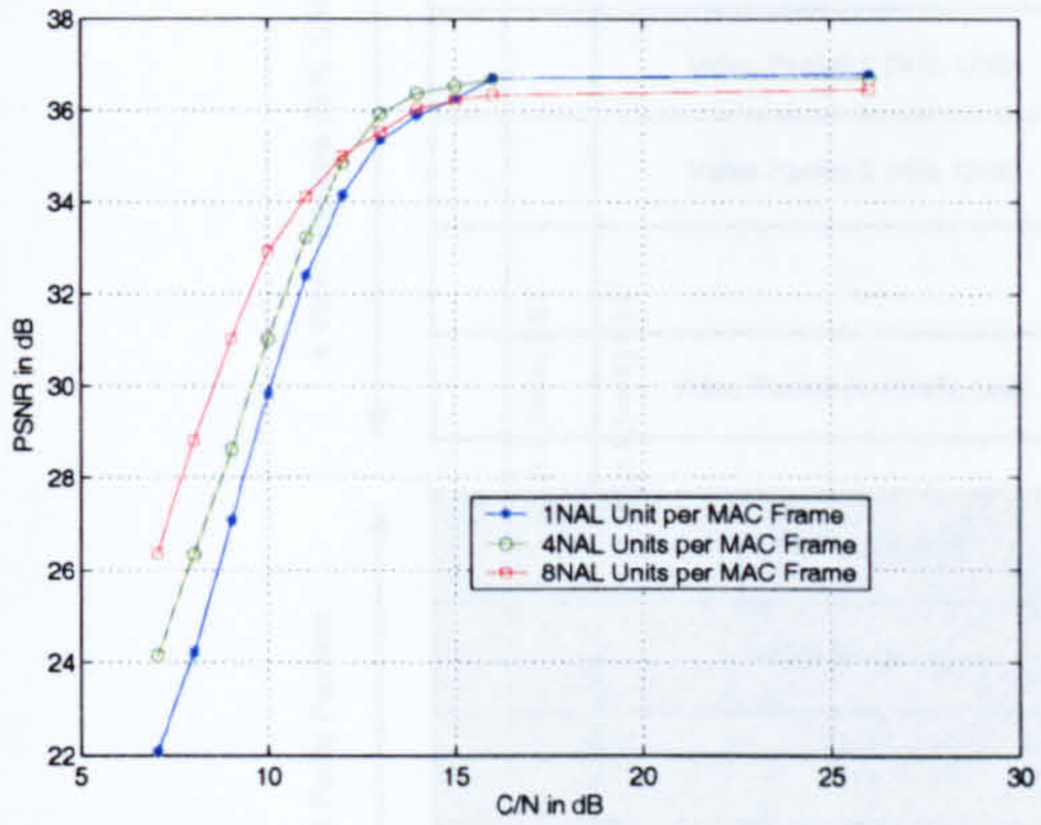


(a) Different Number of NALUs per MAC frame

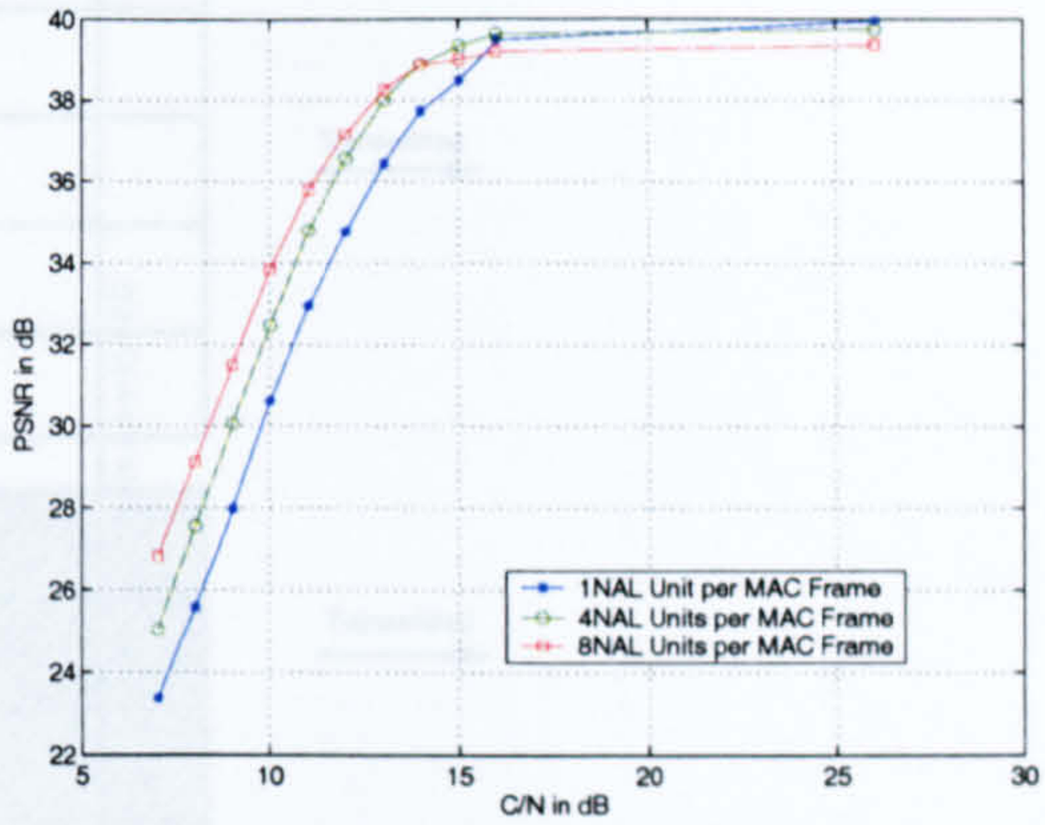


(b) Different Power Levels

Figure 6.17: PSNR Comparison for Proposed MAC, Foreman at 128kbs, Fixed PHY packet size, no ARQ (Broadcast)



(a) 512 kbits/s



(b) 1024 kbits/s

Figure 6.18: PSNR Comparison for Proposed MAC, Foreman, Fixed PHY packet size versus received power for different number of NAL Units, with no ARQ (Broadcast)

6.7 Cross-Packet Forward Error Correction

6.7.1 Principles

Further improvements can be obtained through the addition of Forward Error Correction (FEC) at the application layer [10], in the form of shortened Reed Solomon (RS) codes with erasure decoding [50]. The legacy IEEE 802.11 MAC without aggregation is used in this section. The RS code should be applied across NAL units since with the IEEE 802.11 legacy MAC, erroneous MAC frames are discarded. Moreover, in poor QoS networks, lost IP packets may occur. If RS is applied within a single video packet at the application layer, then the erroneous packet would not be available for error correction since the MAC will drop corrupted frames [50] [134]. MAC-level FEC is under review in the IEEE 802.11 Task Group e [149] [146]. Chapter 5 however showed that packets could get heavily corrupted and FEC on a single packet basis might not be an appropriate solution if applied at the MAC level.

To take advantage of the missing packets, a RS code with erasure decoding should be applied across NAL units at the application layer, as shown in figure 6.19, so that missing NAL units can be recovered [76, 134]. For cross-packet FEC at the application layer, NAL units must all have the same size L_N. To achieve fixed packet length, padding can be used, at the expense of around 1-2 % of overhead. The FEC system presented uses a *RS(n,k)* error correction and erasure scheme with a packet interleaver of depth n.

As shown in figure 6.19, k data packets of length L_N are buffered in the interleaver at the application level and n-k parity packets are generated by the cross-packet RS encoder [134]. The interleaver is only used to generate the parity packets. Data packets are transmitted normally, i.e. non interleaved. The RS decoder can recover the k data packets if a maximum of t=n-k (correction capability) packets are missing [12, 139]. The k data packets can therefore be reconstructed if any k are received out of the n

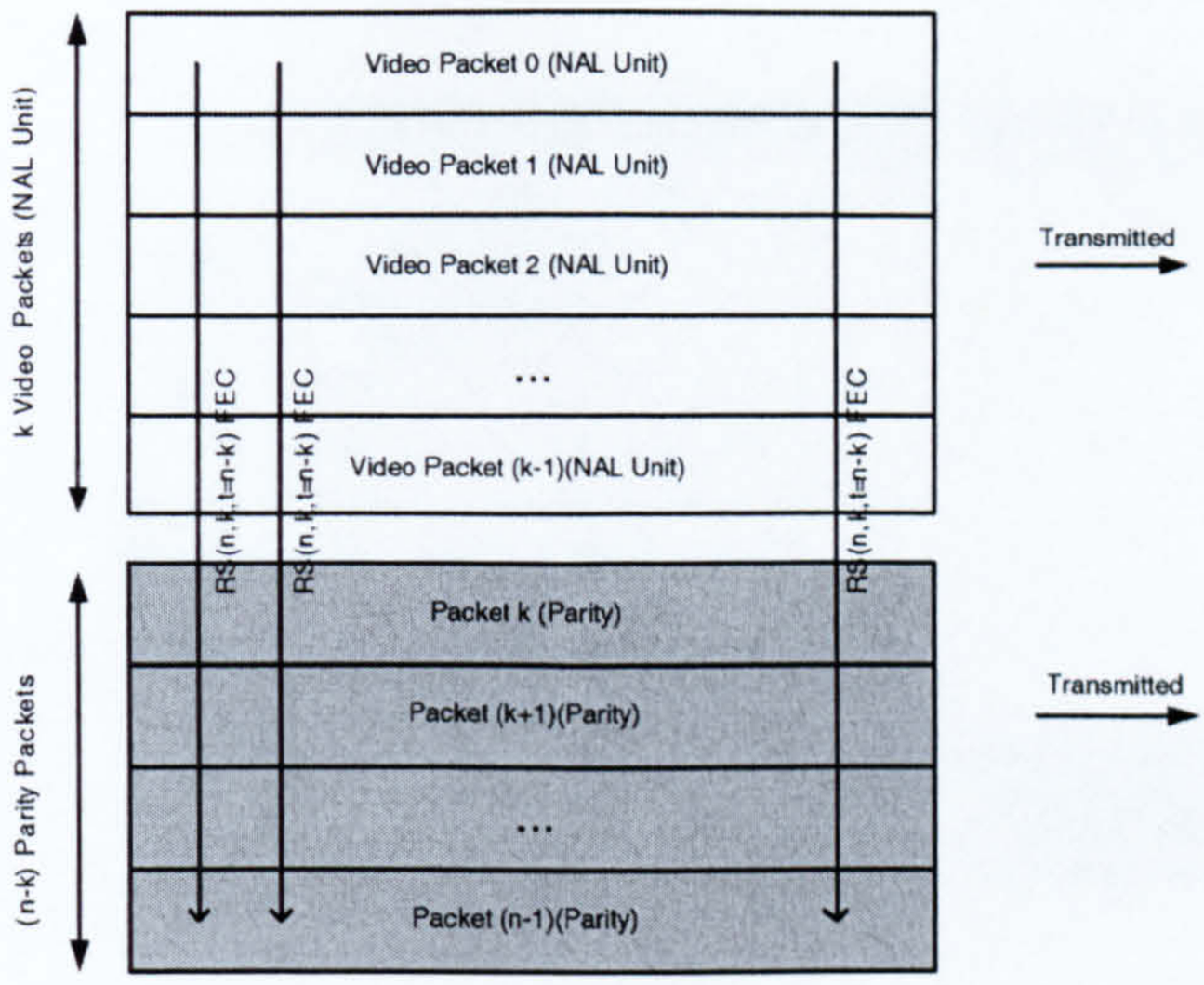


Figure 6.19: Cross-Packet FEC

transmitted packets. The main considerations in the choice of n and k are ([12]):

- Encoding: the interleaver depth n determines the encoding delay and the encoding complexity.
- Decoding: in the event of packet loss, the decoder has to wait at least k packets before decoding can be performed. In order to minimize the decoding delay, the value of k should be small.
- The coding rate $\frac{k}{n}$ determines the amount of overhead.
- The correction capability t=n-k determines the robustness of the code to burst losses.

The correction capability, coding rate and interleaver depth are closely linked as shown in figure 6.20. For example, for a given correction capability of 2 packets, an interleaver of depth 8 leads to a coding rate of $6/8 = 0.75$ with 25% overhead. An interleaver depth of 16 leads to a coding rate of $14/16 = 0.875$ with a 12% overhead.

6.7.2 Results

In this section, the channel resources and the encoding parameters are fixed. Focus is given to the broadcast transmission, i.e. where no ARQ at any level can be used. The MAC payload size is fixed at 752 bytes and a single NAL unit (plus RTP/UDP/IP header) is mapped into each MAC frame. The PER is therefore equal to the NER. This section compares a standard H.264 transmission (without FEC) with the proposed cross-packet FEC. The parameters varied in this study include: the interleaver depth (8, 16, 32 and 64), the correction capability (1, 2, 4, 8 and 16 packets) and the coding rate (1 - no FEC, 0.875 and 0.75). For a fixed transmission rate, and given the overheads introduced by the FEC, the encoded video bit rate drops from 715 kbits/s with no

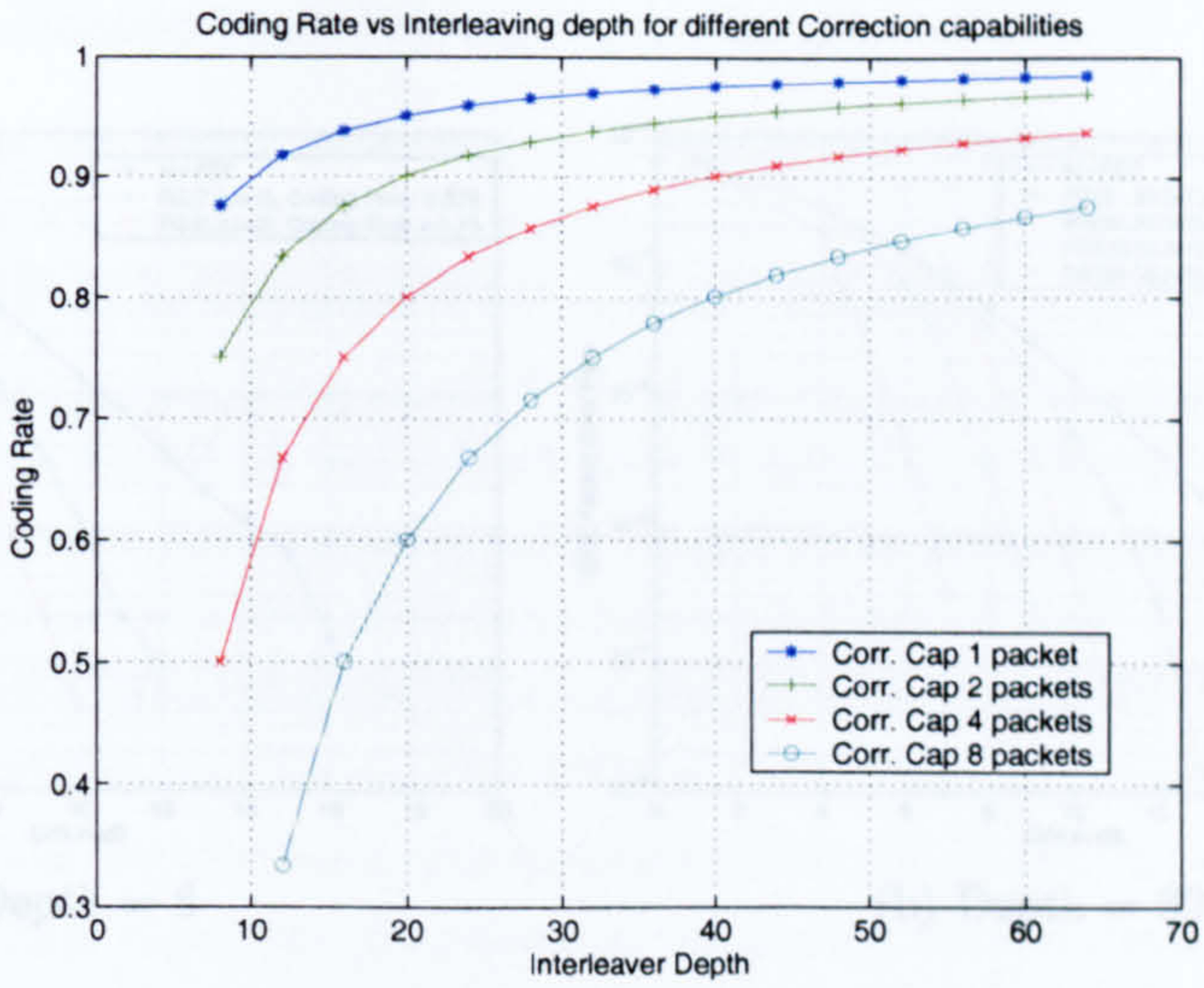


Figure 6.20: Influence of the Interleaver depth

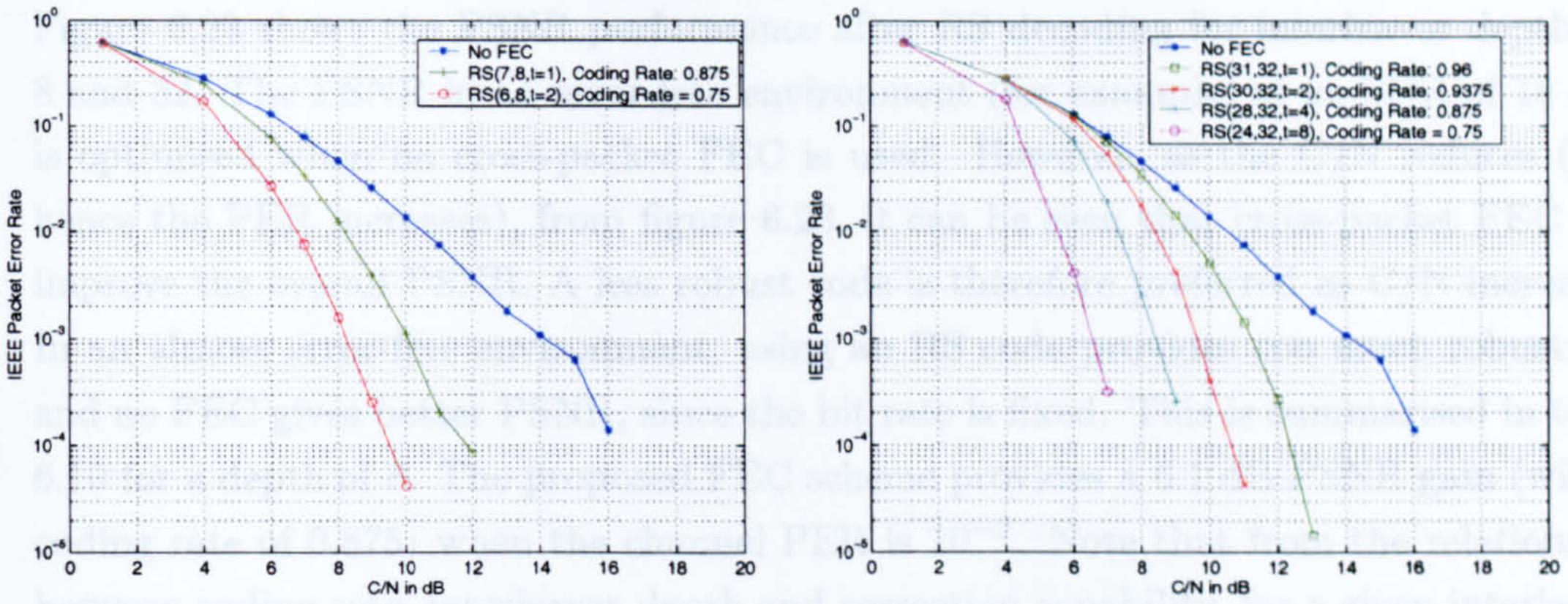
FEC, to 625 kbits/s for a coding rate of 0.875, and 536 kbits/s for a coding rate of 0.75, as shown in table 6.8.

Table 6.8: FEC Comparison

Coding Rate	*Overhead in %*	*Video Bit Rate*	*Encoded PSNR*
1 (No FEC)	0	715 kbit/s	37.92 dB
0.875	12.5	625 kbit/s	37.66 dB
0.75	25	536 kbit/s	36.93 dB

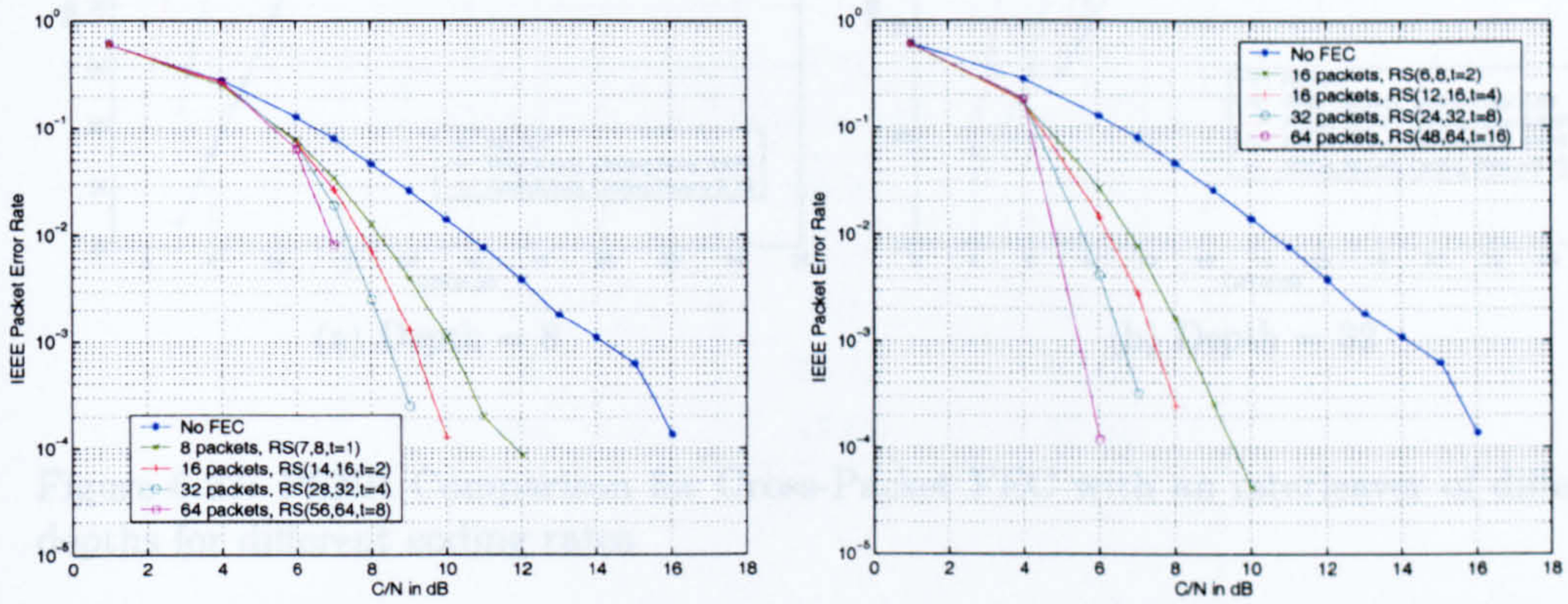
6.7.2.1 Cross-Packet FEC PER Performance

Figure 6.21 shows the PER improvements after RS decoding for interleaver depths of 8 and 32 and for different coding ratios (and therefore different correction capabilities). As expected, the use of RS enhances the PER performance. With a depth of 8, a coding rate of 0.875 offers a 2.25dB gain for a PER of 10^{-2}, whereas a coding rate of 0.75 offers 3.75 dB of improvement. Moreover, for a given coding rate, the depth of the interleaver influences the performance as shown in figure 6.22. With a depth of 8, a coding rate of 0.75 offers 3.75dB gain for a PER of 10^{-2}, whereas a depth of 32 offers 5dB of improvement. This is explained by the fact that for a given coding rate (0.875), with the depth increasing from 8 to 32, the correction capability goes from 1 packet to 4 packets. RS(7,8,t=1) can recover the original packets when 1 packet out of 8 is missing, whereas RS(28,32,t=4) can recover the original packets when 4 packets out of 32 are missing. For a given coding rate, the larger the depth, the more robust the RS code. The C/N gains of the PER performance for a PER of 10^{-2} and 10^{-3} for coding ratios of 0.875 and 0.75 and for depths of 8,16 32 and 64 are summarised in table 6.9.



(a) Depth = 8 (b) Depth = 32

Figure 6.21: PER Comparison for Cross-Packet FEC with an interleaver with different depths and for different coding rates



(a) Coding rate = 0.875 (b) Coding rate = 0.75

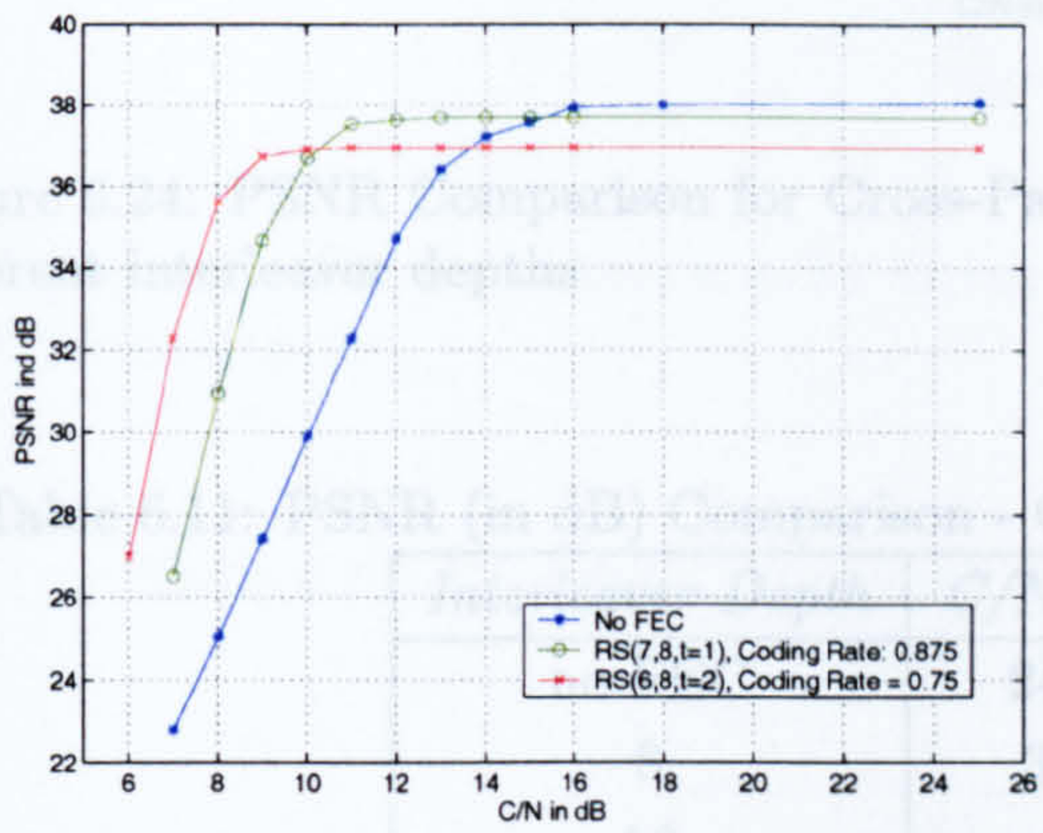
Figure 6.22: PER Comparison for Cross-Packet FEC with an interleaver with different depths and for different coding rates

Table 6.9: PER Performance - C/N Gains over No FEC

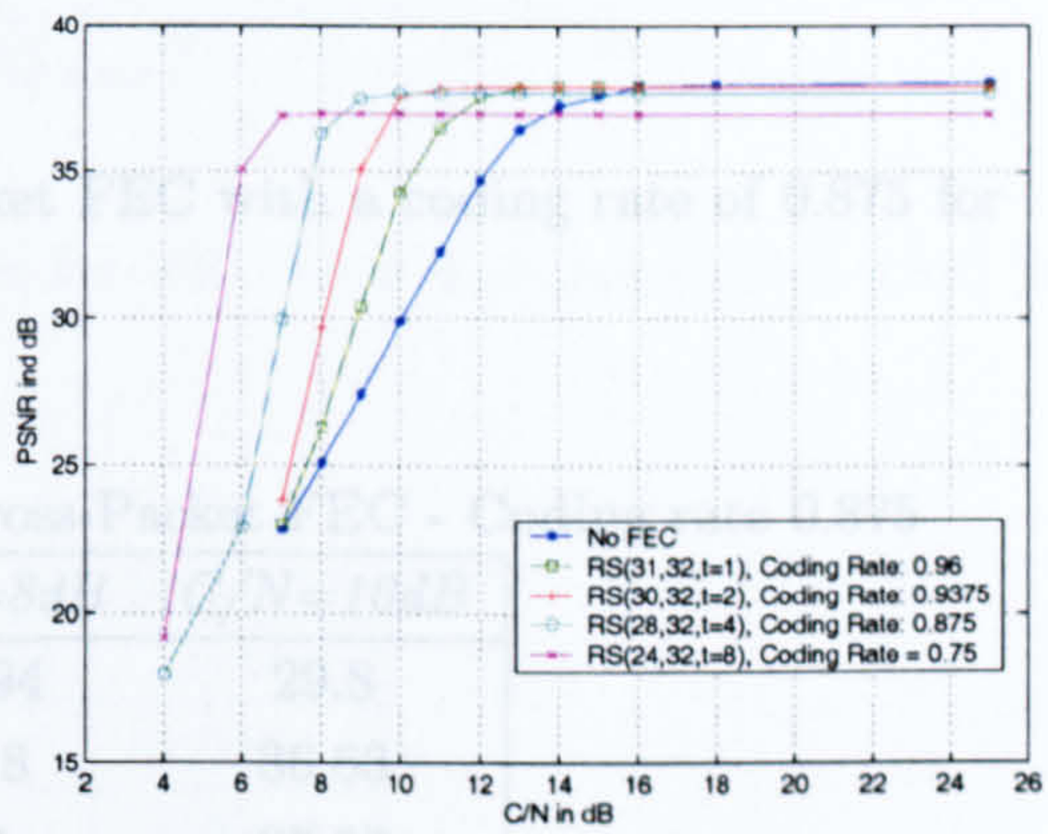
Interleaver depth	*Coding Rate* $PER = 10^{-2}$	*0.875* $PER = 10^{-3}$	*Coding Rate* $PER = 10^{-2}$	*0.75* $PER = 10^{-2}$
8	2.25 dB	4 dB	3.75 dB	5.8 dB
16	2.75 dB	5 dB	4.25 dB	6.7 dB
32	3.2 dB	5.7 dB	5 dB	7.4 dB
64	3.6 dB	6.6 dB	5.7 dB	8.6 dB

6.7.2.2 Cross-Packet FEC PSNR Performance

Figure 6.23 shows the PSNR performance after RS decoding for interleaver depths of 8 and 32. The PSNR in an error-free environment (for example, at an SNR of 18 dB) is optimised when no cross-packet FEC is used. However, as the C/N reduces (and hence the PER increases), from figure 6.23, it can be seen that cross-packet FEC can improve the overall PSNR. A less robust code is therefore preferred as C/N increases. In an almost error-free environment, using an RS code provides too much robustness and no FEC gives better PSNR, since the bit rate is fixed. This is summarised in table 6.10 for a depth of 8. The proposed FEC scheme provides a 6.1 dB PSNR gain (with a coding rate of 0.875) when the channel PER is 10^{-2}. Note that from the relationship between coding rate, interleaver depth and correction capability, for a given interleaver depth, the correction capability decreases as the coding rate increases.



(a) Depth = 8



(b) Depth = 32

Figure 6.23: PSNR Comparison for Cross-Packet FEC with an interleaver of different depths for different coding rates

Table 6.10: PSNR (in dB) Comparison - Cross-Packet FEC - Depth of 8

PER on the channel	*C/N (in dB)*	*Coding Rate* 1	*0.875*	*0.75*
0	-	37.92	37.66	36.93
2×10^{-4}	15.75	37.81	37.66	36.93
10^{-3}	14.10	37.21	37.65	36.92
10^{-2}	10.49	30.98	37.08	36.91
10^{-1}	6.46	21.30	23.80	29.23

Figure 6.24 compares the PSNR of the *foreman* sequence for a coding rate of 0.875 and for a number of different interleaver depths. For a given coding rate, the error correction capability is seen to improve with increasing interleaver depth. This explains the better PSNR values seen for higher interleaver depths. Table 6.11 summarises the PSNR improvements over the no FEC case by using different interleaver lengths for a coding rate of 0.875. For a C/N of 8dB, no FEC offers a PSNR of 25dB. Using an

interleaver with a depth of 8 allows a gain of almost 6dB in PSNR.

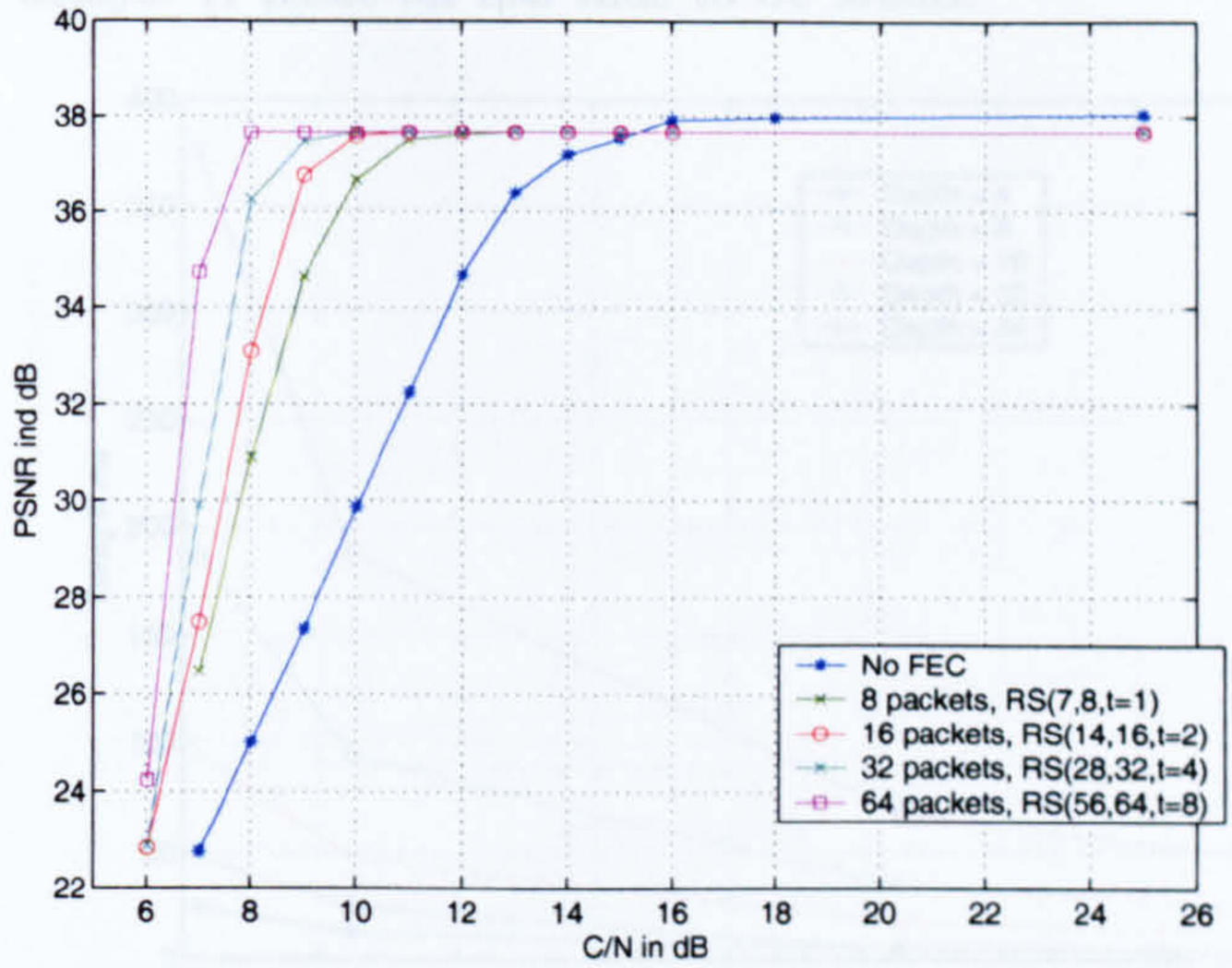


Figure 6.24: PSNR Comparison for Cross-Packet FEC with a coding rate of 0.875 for different interleaver depths

Table 6.11: PSNR (in dB) Comparison - Cross-Packet FEC - Coding rate 0.875

Interleaver Depth	*C/N=8dB*	*C/N=10dB*
no FEC	24.94	29.8
8	30.8	36.63
16	33	37.57
32	36.12	37.66
64	37.66	37.66

6.7.2.3 Cross-Packet FEC Delay Performance

Increasing the depth increases the correction capability for a given coding rate and improves both the PER and PSNR performance. However, it also increases the packetisation delay, as shown in figure 6.25. In order to deliver the received packets to the video decoder, they are buffered prior to RS decoding. To deliver the first packet from this buffer, the RS decoder must wait for at least k packets to be received and processed. The delay is video bit rate rate dependent. The faster the data is transmitted, the shorter is the delay for a given depth. This only considers the delay incurred by the cross-packet FEC and the interleaver and does not take into account MAC and transmission delays. This therefore does not represent the end-to-end system delay. At low video bit rates, delays can exceed 50 ms if the depth is higher than 4, which can not be acceptable. As the video bit rate increases, the delay decreases. However, for a depth of 64, the delay is always larger than 20 ms for video bit rates up to 1 Mbits/s. Table 6.12 summarises the delay with a coding rate of 0.875 at a video bit rate of 625kbits/s for different depths. A depth of 16 presents a delay of 17 ms in this case. Using a large

depth might increase the robustness and the PER and PSNR performance, however, it presents large delays. A trade-off has thus to be found.

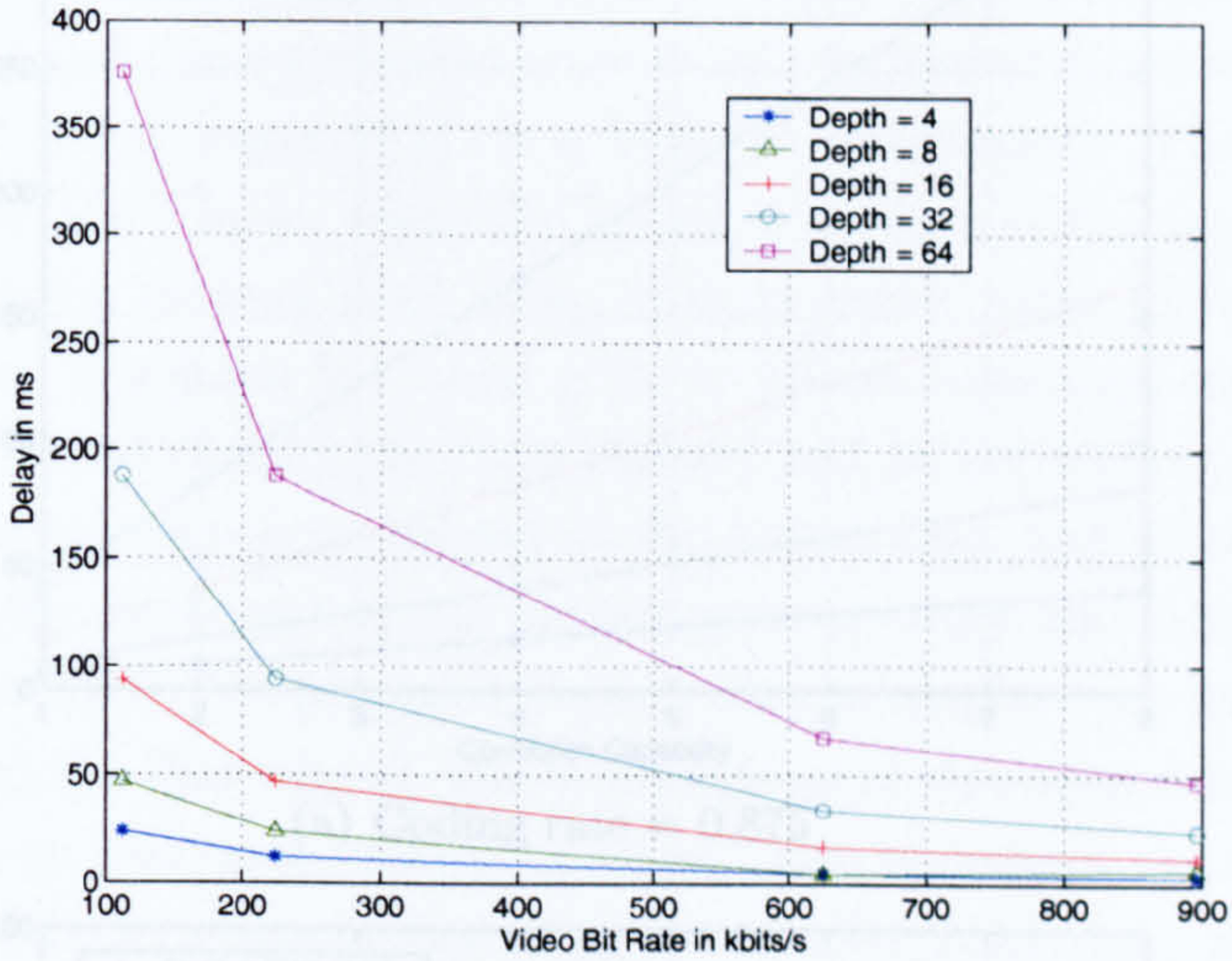
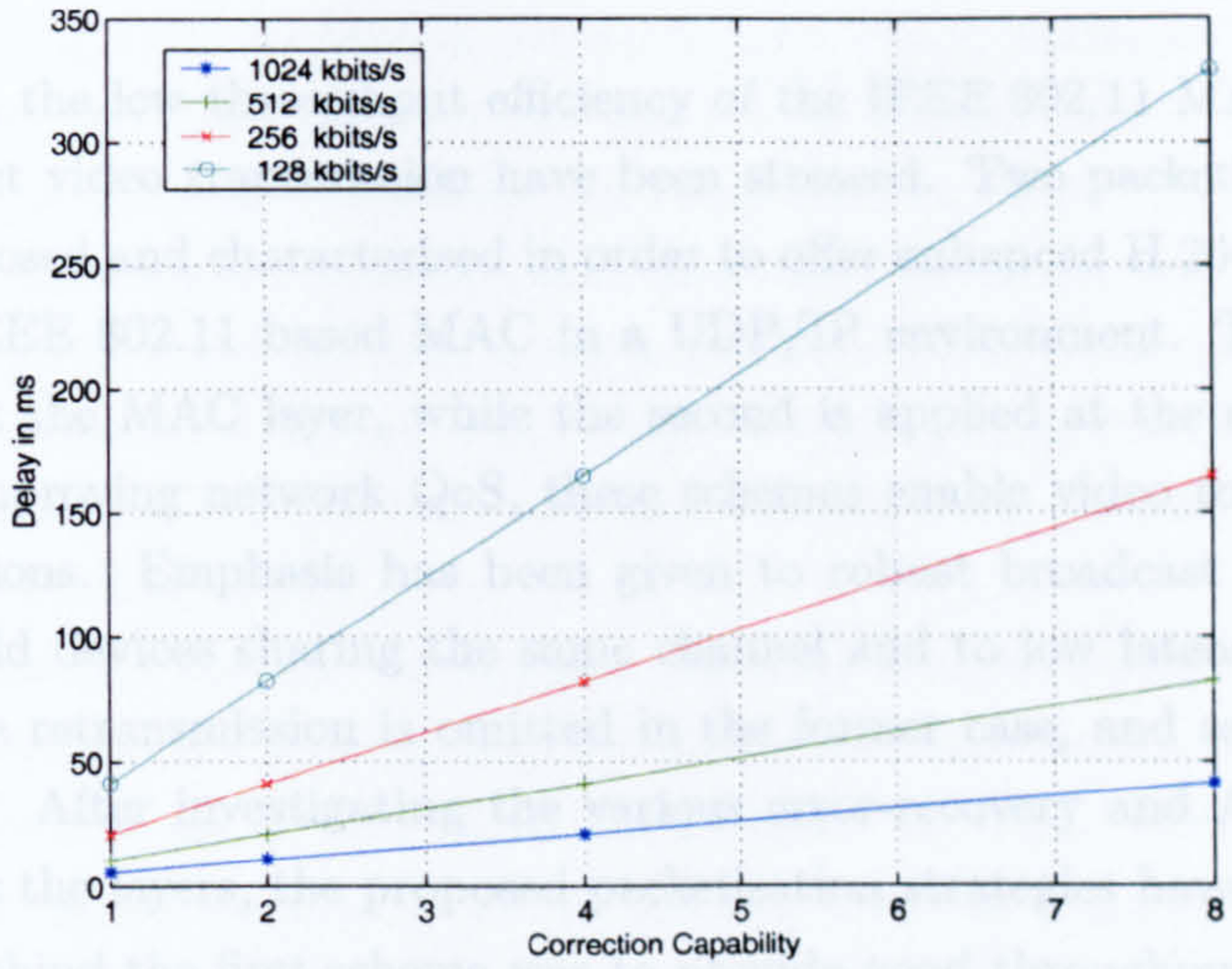


Figure 6.25: Delay Comparison for Cross-Packet FEC with a coding rate of 0.875 for different depths

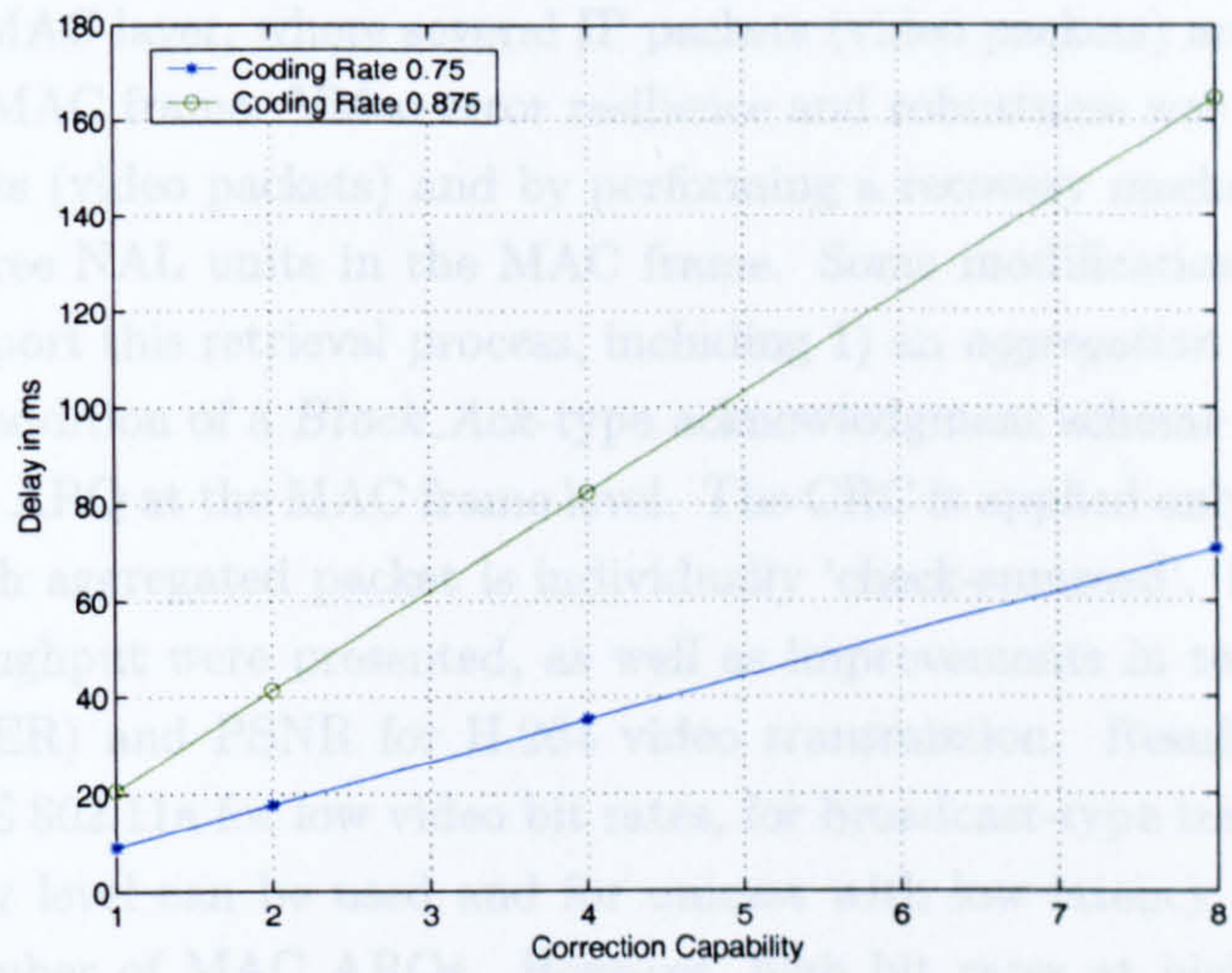
Table 6.12: RS delay Comparison for a coding rate of 0.875 at 625kbits/s

Interleaver Depth	*Delay*
4	4.20ms
8	8.40ms
16	16.82ms
32	33.64 ms
64	67.28 ms

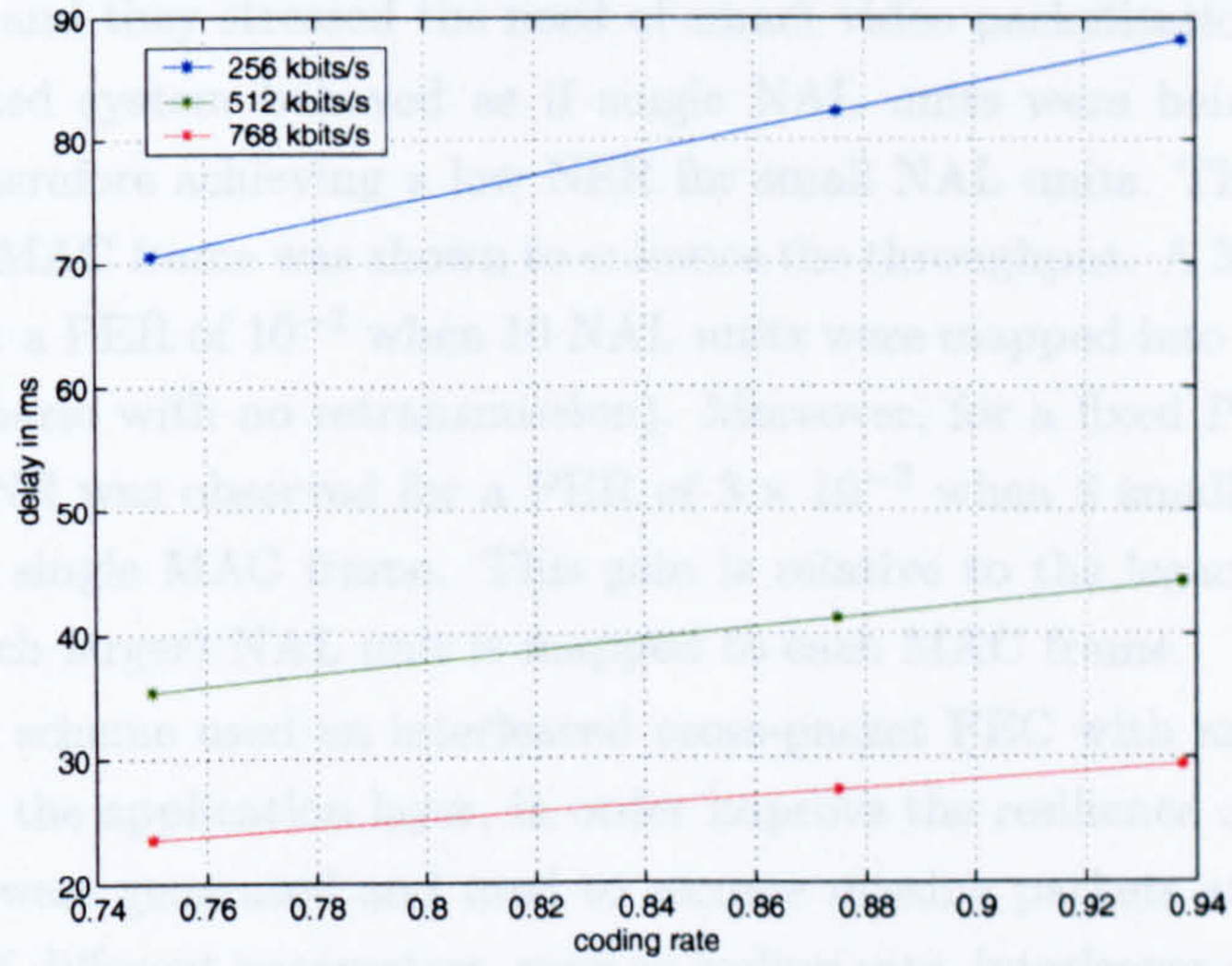
Figure 6.26 compares the delay for different correction capabilities and different coding rates. It can be seen that as the correction capability increases, the delay also increases. This is explained by the fact that for a given coding rate, a larger correction capability requires a larger depth. Nevertheless, for a given depth, decreasing the coding rate increases the overheads but decreases also the delay since fewer packets are required to perform packet recovery with a lower coding rate, as shown in figure 6.26(c) for a depth of 32.



(a) Coding rate = 0.875



(b) Video Bit Rate = 256 kbits/s



(c) Depth 32

Figure 6.26: Delay Comparison for different Correction capabilities

6.8 Conclusions

In this chapter, the low throughput efficiency of the IEEE 802.11 MAC layer and the need of a robust video transmission have been stressed. Two packetisation strategies have been proposed and characterised in order to offer enhanced H.264 video transmission over an IEEE 802.11 based MAC in a UDP/IP environment. The first strategy is performed at the MAC layer, while the second is applied at the application layer. Rather than improving network QoS, these schemes enable video to operate in poor channel conditions. Emphasis has been given to robust broadcast transmissions to several handheld devices sharing the same channel and to low latency unicast transmissions, where retransmission is omitted in the former case, and severely limited in the latter case. After investigating the various error-recovery and ARQ mechanisms available across the layers, the proposed packetisation strategies have been discussed.

The idea behind the first scheme was to provide good throughput for video transmission, as well as to guarantee video robustness. This is realised via an aggregation scheme at the MAC layer, where several IP packets (video packets) are aggregated into a single larger MAC frame. Video error resilience and robustness was ensured by using small NAL units (video packets) and by performing a recovery mechanism in order to retrieve error-free NAL units in the MAC frame. Some modifications were necessary in order to support this retrieval process, including 1) an *aggregation field* in the MAC header, 2) the addition of a *Block_Ack*-type acknowledgment scheme and 3) the use of selective repeat ARQ at the MAC frame level. The CRC is applied only across the MAC header and each aggregated packet is individually 'check-summed'. Enhancements to the MAC throughput were presented, as well as improvements in terms of NAL unit Error Rate (NER) and PSNR for H.264 video transmission. Results were given for mode 1 of IEEE 802.11a for low video bit rates, for broadcast-type transmission, where no ARQ at any level can be used and for unicast with low latency transmission, i.e. with a low number of MAC ARQs. However, high bit rates at higher modes would provide similar results. Moreover, results are also representative of larger numbers of retransmission and they stressed the need of smart video packetisation strategies.

The proposed system behaved as if single NAL units were being independently transmitted, therefore achieving a low NER for small NAL units. The use of multiple NAL units per MAC frame was shown to enhance the throughput. A 3dB gain in PSNR was reported at a PER of 10^{-2} when 10 NAL units were mapped into each MAC frame (broadcast scenario with no retransmission). Moreover, for a fixed PHY length, a 2.5 dB gain in PSNR was observed for a PER of 3×10^{-2} when 8 small NAL units were mapped into a single MAC frame. This gain is relative to the legacy MAC, where a single (and much larger) NAL unit is mapped to each MAC frame.

The second scheme used an interleaved cross-packet FEC with an RS erasure correction code at the application layer, in order improve the resilience of the system. RS parity packets were generated and used to recover missing packets at the application. The influence of different parameters, such as coding rate, interleaver depth and correction capability on the received quality, as well as delay, were discussed for the specific

case of broadcast transmission.

For a fixed coding rate, increasing the interleaver depth enhances the PER and PSNR performance, but also increases the delay. For a fixed depth, decreasing the coding rate enhances the PER, improves the robustness (a better PSNR is achieved in an error-prone transmission channel) and decreases the delay, but also increases overheads (with a fixed video bit rate, a worse PSNR is achieved in an almost error-free environment compared to the no FEC case). A 6 dB gain in PSNR was observed using the cross-packet FEC (0.875 code rate and a depth of 8), compared to standard packetisation, at a PER of 10^{-2}.

In order to overcome the low throughput efficiency without losing visual quality, the first proposed system is a possible solution for robust video transmission and other multimedia services over an IP and IEEE 802.11-based network. Moreover, an optimisation algorithm would be able to derive the best suitable mapping configuration for given channel conditions. The use of cross-packet FEC seems suited for broadcast usage. A combined scheme using multiple NAL units mapped into a single MAC frame with cross-packet FEC is possible, and this would further enhance system performance.

Chapter 7

Investigation of a Cross Layer Link Adaptation Mechanism for Video Transmission over Wireless LANs

7.1 Introduction

Numerous transmission techniques are available on a protocol stack and a wide range of optimisations can be found in the literature. Much of the research over the last decade has focused on the enhancement of each particular layer without considering adjacent layers. It is however preferable to improve one specific layer with the knowledge of its impact on the overall system performance [50]. More specifically, layers may coordinate their efforts and share information in order to enhance the system performance.

Adapting the video coding to the channel/network conditions and technologies (and *vice versa*) via the cross-layer exchange of information has not been extensively investigated until recent years. In [53], the authors describe channel-adaptive video streaming technologies. These include an Adaptive Media Play-out (AMP) in order to smooth out and reduce jitter and delay, an optimised packet scheduler with rate distortion in order to allocate resources and bandwidth among packets depending on the networks resources. In [160], the authors develop the idea of *inter-layer* communications for multimedia delivery over 3G and 4G with the *all IP* concept. The Joint source-channel coding paradigm for QoS support over 4G is investigated, combining transport techniques, such as UDP-lite [155] and robust header compression (RoHC) [161] with channel diversity, entropy coding, error resilience and decoding techniques. In [50], the authors develop a cross-layer optimisation, combining application layer FEC, adaptive MAC retransmission and adaptive video packet size for the case of *Fine Granularity Scalability* (FGS) video transmission over an IEEE 802.11b network. In [162], the authors discuss the challenges and principles of a cross-layer optimised multimedia transmission. The paper defines the various options to tackle the cross-layer optimisa-

tion (from the application layer at the top to the bottom, or from the PHY layer at the bottom to the top, or with a application-centric or MAC-centric approach). The choice of optimal modulation using Application/MAC/PHY interactions for FGS video over an IEEE 802.11b network is discussed as well as the choice of a modulation scheme for optimal power consumption. Moreover, the authors stress the fact that an optimal solution for throughput may not be appropriate for multimedia transmission.

In [55], the author details the basis of a cross-layer framework where information is exchanged between layers. Their study allows the upper layer to better adapt their strategies to varying link and network conditions. This includes dynamic packet size adaptation, for a given link layer and channel condition, and, for a given packet length, the scheme optimises the link layer parameters, such as constellation and symbol rate, in order to optimise the throughput. Their study also investigates the joint allocation of capacity and flow, as well as smart packet scheduling and rate allocation in order to overcome network-congestion by assigning transmission priority to packets. However, apart from [55] and [162], adaptive link and MAC layer techniques, such as coding rate and modulation scheme have rarely been considered in the design of cross-layer systems.

In chapter 2, the PHY layer of COFDM-based WLANs at 2.4GHz and 5GHz was studied. Numerous modes are available, each providing different throughput and reliability levels. Table 7.1 recalls the different operating modes available for the IEEE 802.11a/g [3, 4] PHY layer. They range from BPSK 1/2 rate (mode 1) which provides a nominal bit rate of 6 Mbits/s to 64 QAM 3/4 rate (mode 7), with a nominal bit rate of 54 Mbits/s. The BPSK 1/2 rate mode provides a more reliable transmission link than the 64 QAM 3/4 rate mode for a given receive power. Figure 7.1 recalls the PER performance of the 7 main modes of IEEE 802.11a/g. A Carrier to Noise Ratio (C/N) of 15dB provides error free transmission if mode 1 is used, whereas mode 7 presents a PER of 0.9 at this C/N value. The choice of the operating mode is therefore crucial to system performance.

Table 7.1: Mode Dependent Parameters for IEEE802.11a/g

Mode	*Modulation*	*Coding rate*	*Nominal bit rate (Mbit/s)*
1	BPSK	1/2	6
2	BPSK	3/4	9
3	QPSK	1/2	12
4	QPSK	3/4	18
5	16QAM (IEEE only)	1/2	24
6	16QAM	3/4	36
7	64QAM	3/4	54
8	64QAM (IEEE only)	2/3	48

Due to the numerous operating modes available at the PHY layer, each with their own unique characteristics, the ability for the system to adapt to the fluctuations of

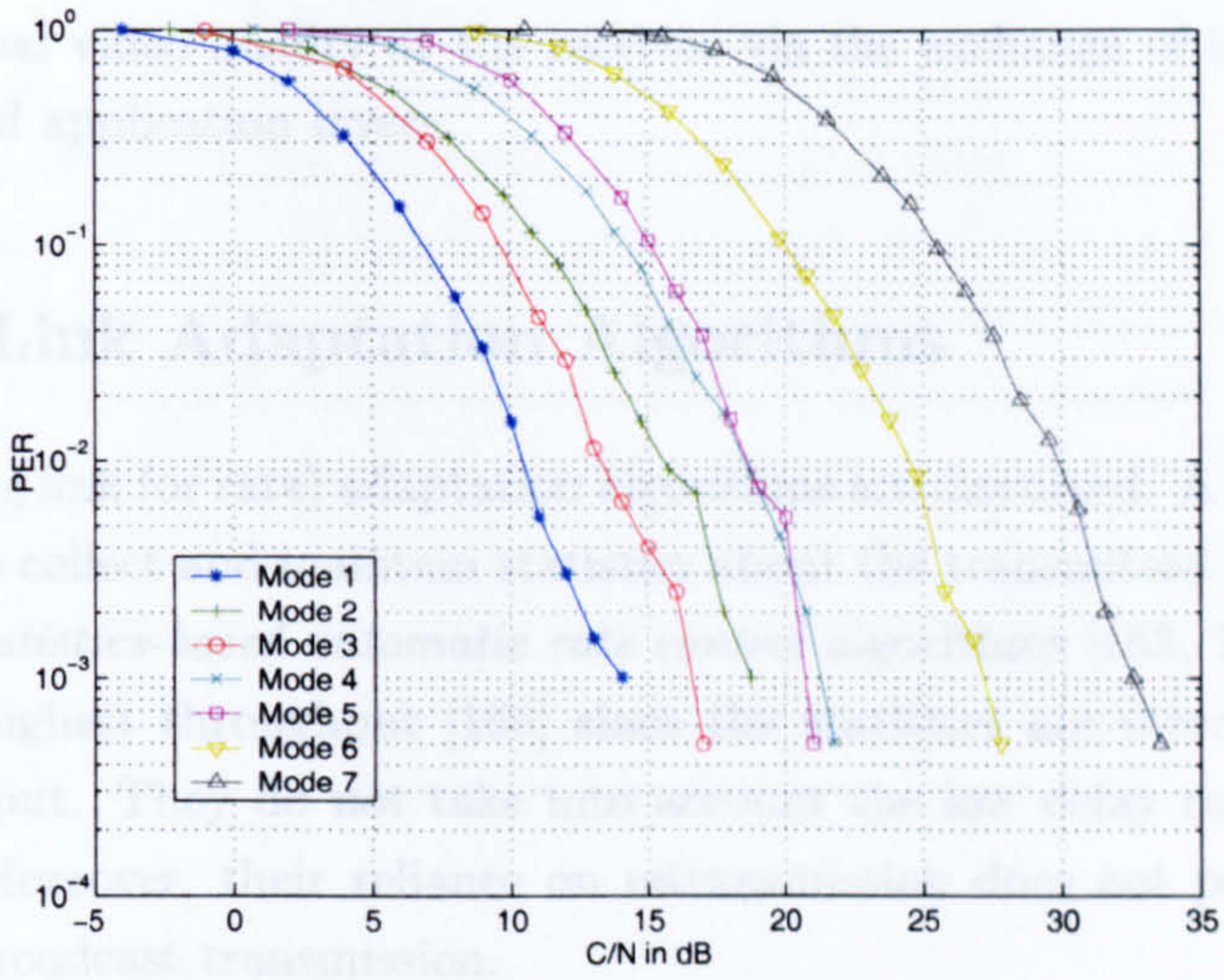


Figure 7.1: IEEE 802.11a/g Characteristics. PER curves - ETSI-Bran Channel A-model - 188 byte packets

the environment (mobility, interference and congestion) is critical to ensure overall performance optimisation. Many parameters can be varied at the MAC and PHY levels, examples include the number of MAC level retries allowed, the packet size, the mode (modulation and coding rate) and the type of antenna. Neither the IEEE 802.11 [39], nor the IEEE802.11a/g standards specify an algorithm for dynamic rate switching. The IEEE 802.11 MAC only defines rules for the mode selection of the management frames and declares the dynamic rate selection beyond the scope of the specifications [39, 163, 164]. It is therefore up to the manufacturers and vendors to implement their own algorithms.

The link adaptation mechanism enables the system to adapt the transmission mode according to a quality metric. The ability to change modes is used to control the reliability of the system and provides the radio with the capability to adapt to a better configuration to improve the QoS of the transmission. Common link adaptation algorithms have mainly focused on maximising the error-free throughput at the higher application layer [165]. These do not take into account the nature of the application data. Moreover, they strongly rely on the use of retransmission and therefore do not take into account transmission delays. Real-time video applications are time-bounded and require a strictly low latency transmission. In addition, completely error-free communication is not essential, especially if robust video compression techniques are applied. In such scenarios, improved decoded video quality can be obtained with a video stream transmitted at a higher bit-rate, but with some degree of transmission error, rather than an error free video stream at a lower bit-rate, as long as the level of errors allows the codec to lie within its operating range. Existing algorithms are not designed, and are, most of the time, not suitable for low delay video applications.

This chapter investigates heuristic link adaptation mechanisms appropriate for the transmission of real-time video. Simulations are performed in order to evaluate the

viability of such mechanisms as well as to present a specific algorithm designed to optimise the perceptual video quality at the receiver via the exchange of information between the MAC and application layers.

7.2 Existing Link Adaptation Algorithms

In this section, existing link (or rate) adaptation algorithms are discussed. A simple way to adapt the rate is to collect and maintain statistics about the transmitted data. Such schemes are called *Statistics-based automatic rate control* algorithms [163, 164]. These aim to provide the highest throughput [166] since the statistics are directly related to user-level throughput. They do not take into account the low delay requirements of the application. Moreover, their reliance on retransmission does not permit their implementation for broadcast transmission.

Note that in this chapter, packet collision is not considered. This assumption is made because a collision is representative of the network status (congestion), rather than the PHY layer conditions. Down-scaling the modes would not improve the QoS of the transmission if two stations collide, i.e. with random backoffs equal to zero at the same time (see chapter 3). The rate adaptation algorithm should not therefore be influenced by collision and congestion statistics.

7.2.1 Throughput based Rate Control

In this algorithm, a constant (small) fraction of data (up to 10%) is sent at two adjacent mode rates (lower and higher than the current rate). At the end of a decision window, the transmitter computes the different throughputs and the switch is made to the rate that provides the highest throughput. In order to have meaningful statistics, the decision window must be long enough (about one second, based on [164]).

7.2.2 Packet Error Rate based Control

In this algorithm, the PER of the transmitted data is used for the selection of the mode. The PER can be determined by counting the ACKs of the IEEE 802.11 MAC frame received at the transmitter during a decision window (a missing ACK means that the corresponding packet has not been received correctly). The PER can be used for down-scaling the rate: if the PER exceeds a threshold, then the current mode is switched to a lower mode. Up-scaling is performed if the PER is below a second threshold and the system then switches to a higher mode. To prevent oscillation between two modes, the system is required to stay on the lower mode after down-scaling for a minimum amount of time. The accuracy of the thresholds will determine the performance and the reactivity of the system [163, 164]. This algorithm has not been initially designed for video transmission and instead it optimises the PER for an improved throughput. It does not take into account the nature of the content and its time-bounded requirements.

7.2.3 Retry-based Control

In these algorithms, the decision metric used is the number of failed ARQs. The decision is therefore made at the transmitter. If a transmission is unsuccessful after a certain number of retries N_{fail}, the mode is down-scaled. It offers a very short response time to channel changes. However, the up-scaling takes longer since the decision is performed with a PER-based control scheme using a decision window. This scheme has been developed under the name of AutoRate Fall Back (ARF) [167, 168] and has been designed to optimise the application throughput [48]. However, if the channel conditions change quickly, the algorithm cannot adapt efficiently, and a short error burst will lead to a long drop in throughput. A simpler version would implement up-scaling after a certain number of successful transmissions in a row, $N_{success}$ [48, 148].

7.2.4 SNR-based Control

In this method, the Carrier to Noise Ratio (C/N) or Signal to Noise Ratio (SNR), which is directly linked to the PER, is used to determine the transmission rate. The throughput can be expressed as a function of the PER and can be estimated as in [26, 44, 74] using:

$$Throughput = R \times (1 - PER) \tag{7.1}$$

where R is the nominal bit rate of the link (see table 7.1). Note that since the link adaptation takes place at the border between the MAC and the PHY, the MAC overheads and the packet length are not taken into account in the calculation of throughput. As explained in chapter 3, the receiver checks the Frame Check Sum (FCS) field in order to detect an error. If the packet is correct, the receiver sends an ACK, otherwise it does nothing. The transmitter therefore knows if a packet is corrupted (no ACK or Timeout). A link adaptation based on throughput is presented in figure 7.2 for a MAC packet length of 188 bytes for the 7 operating modes of the IEEE 802.11a/g PHY layer. For such algorithm, the mode decision is as follows:

For each power level C/N:

$$Mode_T(C/N) = m \in \{M\} \; with \; Throughput(m, C/N) = max_{i \in \{M\}} \{Throughput(i, C/N)\} \tag{7.2}$$

with $M = \{1, 2, 3, 4, 5, 6, 7\}$ and $Mode_T$ being the wining mode.. The crossing points of the curves define the switching points (in terms of C/N) at which the system should up-scale or down-scale.

A simple and straight forward SNR-based algorithm would search in look-up tables available at the MAC for the best throughput for a given C/N. These tables could theoretically be generated prior to any transmission with different packet lengths for all the modes, C/Ns and different channel conditions. It should be noted that the formula assumes that ARQ is used for retransmitting packets until the packet is received correctly, or the maximum number of retries is reached (whatever comes first). Data

are therefore received error free and this method does not take into account the nature of the data or the delays incurred.

It should be noted that the C/N switching levels can be converted into equivalent PHY PER thresholds in the adjacent modes, defining therefore the acceptable level of channel errors before changing mode. This then defines a hybrid throughput-based algorithm with the PHY PER as a metric.

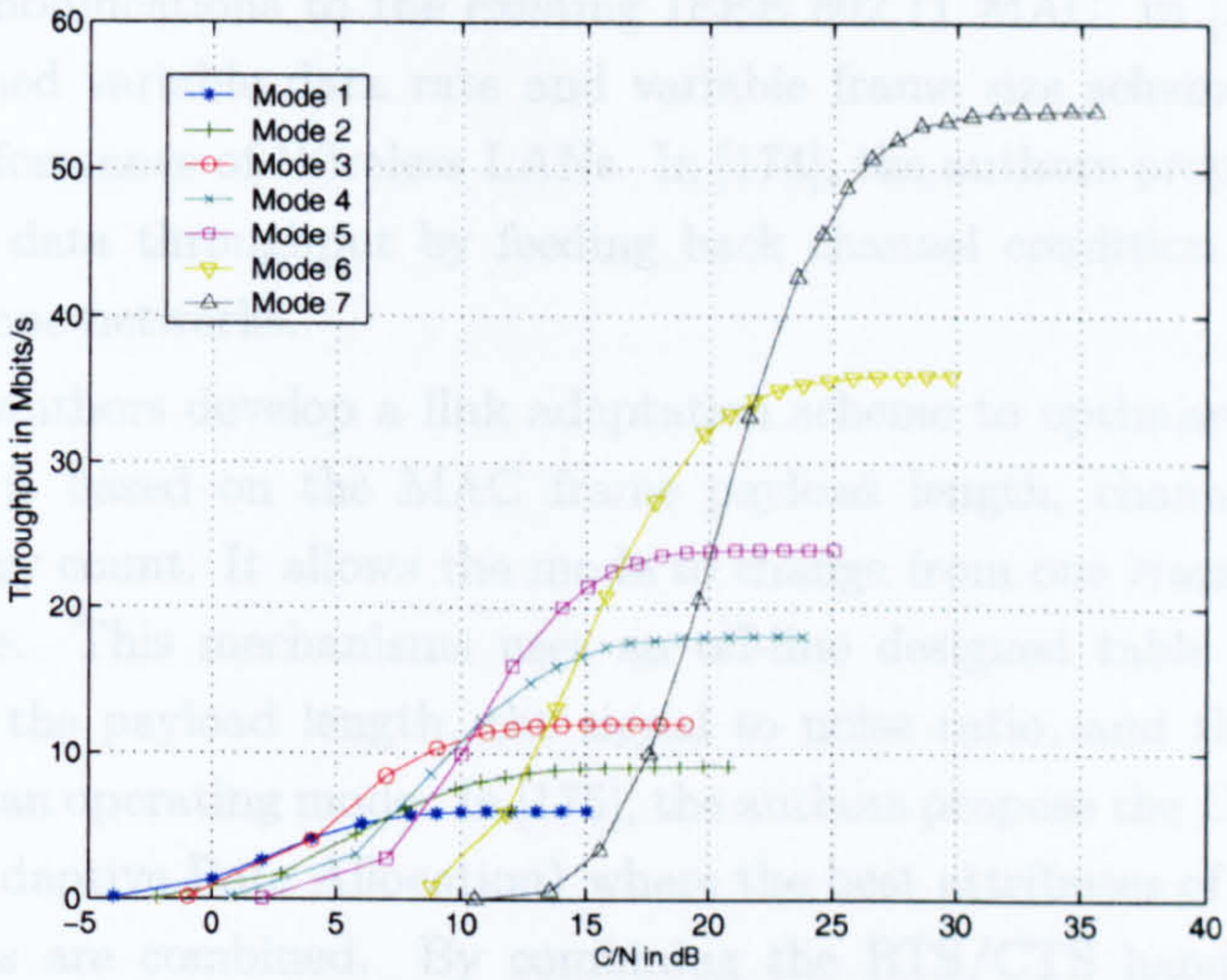


Figure 7.2: Link Adaptation based on Throughput - IEEE 802.11a/g - 188 bytes

7.2.5 Power-based Control

In [169], the authors developed the *MiSer* (Minimum energy transmission Strategy) scheme which minimises the communication energy consumption in an IEEE 802.11a PHY layer extension (IEEE 802.11h) by combining the Transport Power Control (TPC) with the PHY rate adaptation. The set of optimal rate/transmission power pairs is calculated off-line with a specific wireless channel model [48] and the transmitter searches in look-up tables and picks the optimal combinations. This algorithm assumes that the number of stations in the configuration is fixed. It uses the fact that the lower the transmit power (or the higher the PHY rate, i.e. the shorter the transmit duration), the less energy is consumed in one single transmission attempt; the transmission is more likely to fail and will be then be rescheduled, this consuming more energy [169, 170].

In [171], the authors present an algorithm using network-assisted resource management. The scheme enables both link adaptation and power control with EPGRS modulation and coding schemes. An optimal Power Operating Point (OPOP) is defined to manage the power control in addition to the Link Adaptation - Block Error Rate - Switching Point (LA-BLER-SP). A throughput-based control mechanism (using sigmoid and curvature functions) is first applied to choose the mode. The power level is then adjusted to the OPOP.

7.2.6 Other Rate Adaptation Algorithms

Other rate adaptation algorithms combining different techniques have been presented in the literature. In [172], the authors design a Receiver-Based AutoRate (R-BAR) protocol in order to optimise the application throughput [48], where the choice of transmitting rate is made at the receiver based on its own stored statistics [168]. The information on the chosen rate is then transfered back to the transmitter via the CTS frame of the hand-shaking RTS/CTS exchange [148]. The main drawback of this scheme is that it requires some modifications to the existing IEEE 802.11 MAC. In [173], the authors define a combined variable data rate and variable frame size scheme to enhance the throughput performance of Wireless LANs. In [174], the authors propose an algorithm to improve the data throughput by feeding back channel condition estimates at the receiver for ad-hoc networks.

In [22], the authors develop a link adaptation scheme to optimise the throughput. The algorithm is based on the MAC frame payload length, channel condition and MAC frame retry count. It allows the mode to change from one transmission attempt to the next one. This mechanisms uses an off-line designed table with the system status (namely the payload length, the signal to noise ratio, and the retry number) associated with an operating mode. In [175], the authors propose the *CLARA* algorithm (Closed-Loop Adaptive Rate Allocation) where the best attributes of the ARF and R-BAR algorithms are combined. By combining the RTS/CTS handshake with data fragmentation, MAC frame losses and collisions are differentiated. The main point of the scheme is that if a collision occurs, the transmitter would traditionally lower its rate and would therefore lower its throughput. This algorithm uses the exchange of feedback information with the reserved bits in the SERVICE field of the PHY headers.

In [176, 177], the authors develop a hybrid automatic rate controller, combining a throughput-based rate controller with a SNR-based approach. By using RSSI-look up tables, which can be dynamically adjusted, the algorithm selects the more appropriate rate. Not only the throughput is enhanced, the study also aims at reducing the delay and the PER. This work is adopted in [178] where a fast algorithm is designed. The resulting scheme uses the hybrid algorithm to determine the rate switching and then adjusts the transmitted video rate accordingly. A QoS negotiation approach is developed in [179], describing coordination between layers for resource allocations. This algorithm also relies on thresholds that are channel model dependent. In [48], the authors develop an Adaptive ARF (AARF) algorithm. In ARF, typically, the best rate optimising the application throughput is the highest rate whose PER is low enough such that the number of retransmissions is low. But the system also tries to use a higher rate every $N_{success}$ transmissions. Down-scaling occurs after N_{fail} failed transmissions. The system is therefore oscillating when under steady channel conditions since it constantly tries a higher rate. AARF continuously varies $N_{success}$ to reflect changing channel conditions and the system is then able to stabilise.

In [180], the authors define the Opportunistic Auto Rate (OAR) algorithm to improve temporal fairness, where back-to-back packets are transmitted when the channel

quality is considered to be good, in order to better use the channel [148].

It should be noted that the majority of the described link adaptation algorithms aim to improve the throughput [162, 166] and/or reduce the power consumption. They do not take into account the characteristics of the transmitted data. Moreover, in the case of multimedia transmission, they do not optimise the perceived quality [162].

7.3 Link Adaptation based on Video Quality

7.3.1 Motivations

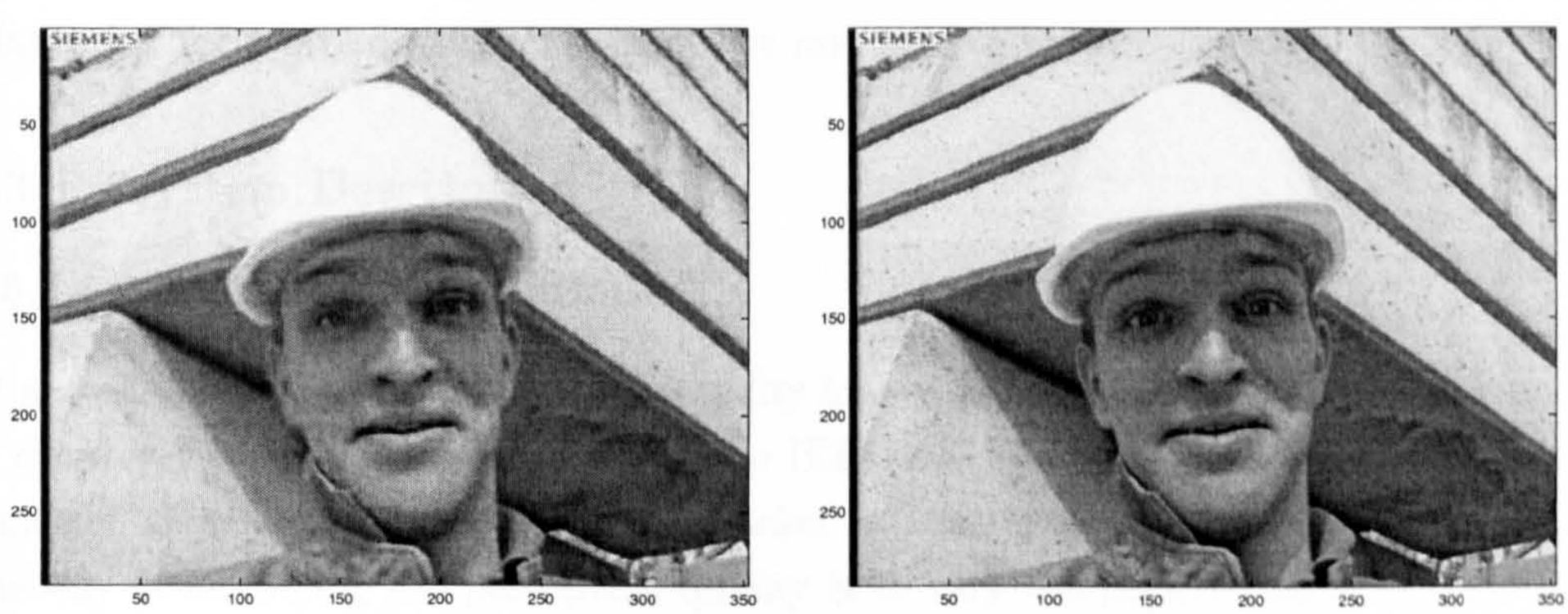
The work in this chapter has been motivated by the fact that the previously presented algorithms do not take into account the nature of the transmitted data. Moreover, they strongly rely on retransmission and cannot therefore be deployed for video applications requiring very low latency. Encoded video should not be transmitted in the same way as normal data. Firstly, the packetisation at the output of the encoder is critical in order to offer robustness and error loss resilience, as shown in chapter 6. Secondly, the transmitted video cannot tolerate excessive delays for display purposes. This only allows a small number of ARQs to be implemented. Thirdly, video encoding exploits temporal redundancy and errors can propagate between frames. Extra care should therefore be made to minimise the resulting visual artifacts, either during the transmission (different levels of protection) or during the decoding and concealment process. The previous link adaptation schemes presented in section 7.2 were designed to provide the highest throughput and to maximise error-free data transfer without regard for delay and retransmission. These schemes implied a high dependency on ARQ at the MAC level and did not take into account the nature of the content. However, for multimedia, a strong reliance an ARQ is not desirable and completely error-free communication is not essential, especially if robust video compression techniques are applied [165]. In such scenarios, better decoded video can be obtained with a video stream transmitted at a higher bit-rate but with some degree of error, rather than an error-free video stream at a lower bit-rate as detailed in table 7.2. The overall quality of the received video sequence depends on a trade-off between video bit-rate and BER/PER as shown in figure 7.3. For a given power level (Carrier to Noise Ratio (C/N) = 18dB), mode 1 provides an error free transmission at low video bit rates (700kbits/s with a PSNR of 37.07dB), whereas mode 5 provides a transmission with a PER of 4×10^{-2} with a higher video bit rate (4235kbits/s). However, figure 7.3(b) shows better resolution and presents a better PSNR (44.85dB) than figure 7.3(a). Impairments due to errors are almost insignificant and cannot be noticed visually thanks to robust concealment [165].

This study is also motivated by the fact that the various operating modes of IEEE 802.11a/g WLANs do not support the same bit rates. For example, High Definition video encoded at 10 Mbits/s cannot be transmitted on the BPSK 1/2 and 3/4 rates, and QPSK 1/2 rate that only allow 6, 9 and 12 Mbits/s at the PHY layer. Limiting the transmission to the higher modes is risky since it reduces the reliability and the coverage of the transmission. It is therefore preferable to adapt the video bit rate

depending on the transmission link.

Table 7.2: Motivation: Video Enhancement

Mode	*Video Bit Rate*	*C/N (dB)*	*PER*	*PSNR (dB)*
1	700 kbits/s	18	0	37.07
5	4235 kbits/s	18	4×10^{-2}	44.85



(a) Mode 1, 700kbits/s, PER = 0, PSNR = 37.07dB

(b) Mode 5, 4235kbits/s, PER = 0.04%, PSNR = 44.85dB

Figure 7.3: *Foreman* Sequence, Frame 30, C/N = 18dB

7.3.2 Scenarios

As defined in section 6.4 of chapter 6, the range of applications where H.264 could operate over a WLAN is very wide. In this chapter, the low number of MAC layer ARQs (0-2) and the relatively low bit rates (250 kbits/s-9000kbits/s) are explained in a similar way to chapter 6, that is real-time live video transmission allowing no retransmission (but with a feedback channel) over several handheld devices sharing the same channel, and low latency unicast transmission without reliance on layer 2 ARQ. The case with one ARQ is however given in order to provide a clear example.

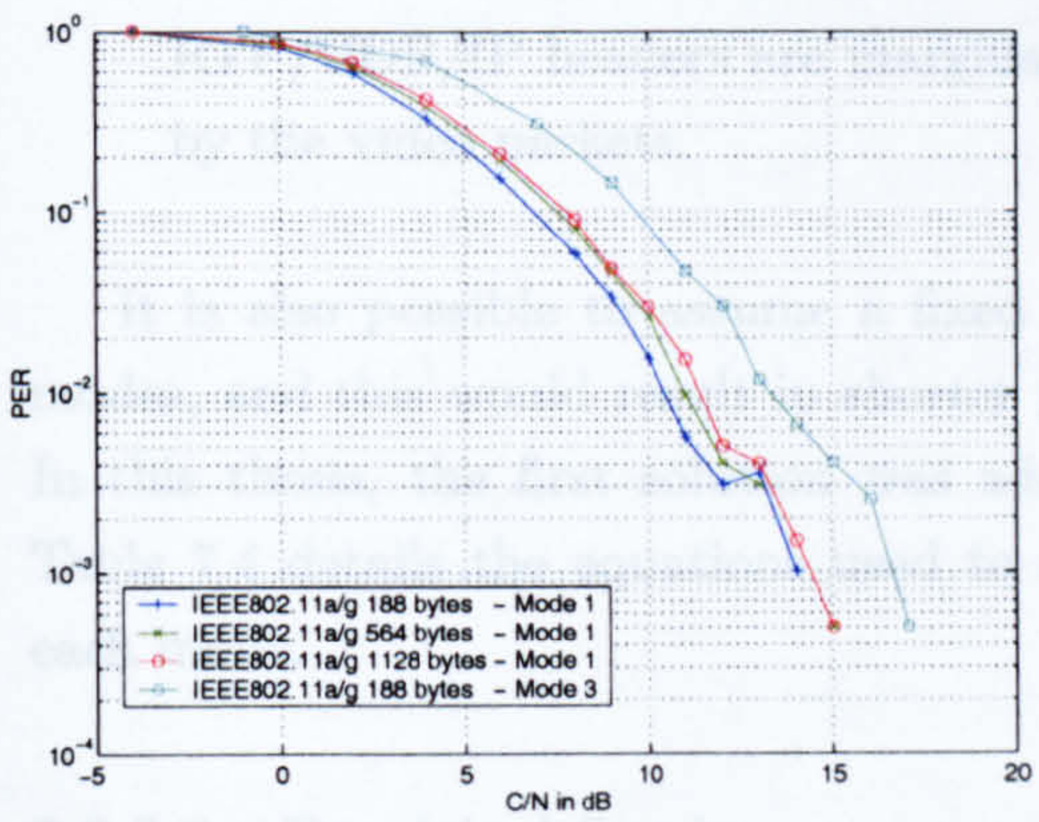
The case of link adaption with a broadcast link is difficult. As far as the author is aware, no cards or access points (APs) available in the market implement link adaptation for broadcast transmissions. There are several reasons for this. Firstly, most of the time, a broadcast link is characterised by the absence of a feedback channel. In IEEE 802.11 networks, this means that the receiver does not acknowledge the received frames. This absence of a feedback channel does not allow the transmitter to compute any form of statistics such as the Received Signal Strength Information (RSSI) or PER. Note, it is however possible to establish a feedback channel in which the receiver can send periodically 'dummy' messages so that the transmitter can compute the RSSI. By assuming a symmetrical channel, the transmitter has therefore knowledge of the channel conditions at the receiver. This solution is however not implemented in practice due to interoperability issues between manufacturers. Secondly, by definition, on a broadcast

line the same data is transmitted transparently to a group of clients. These clients can be in different locations and experiencing different channel conditions, and because the transmission is transparent to the transmitter, it cannot adapt its transmission to each of them. Practically, when used in a broadcast mode, the cards and APs available in the marketplace use the lowest mode (the most reliable and with the widest coverage) to ensure that anyone within range can receive the transmitted data. The case of link adaptation for a broadcast link is therefore not tackled in this thesis.

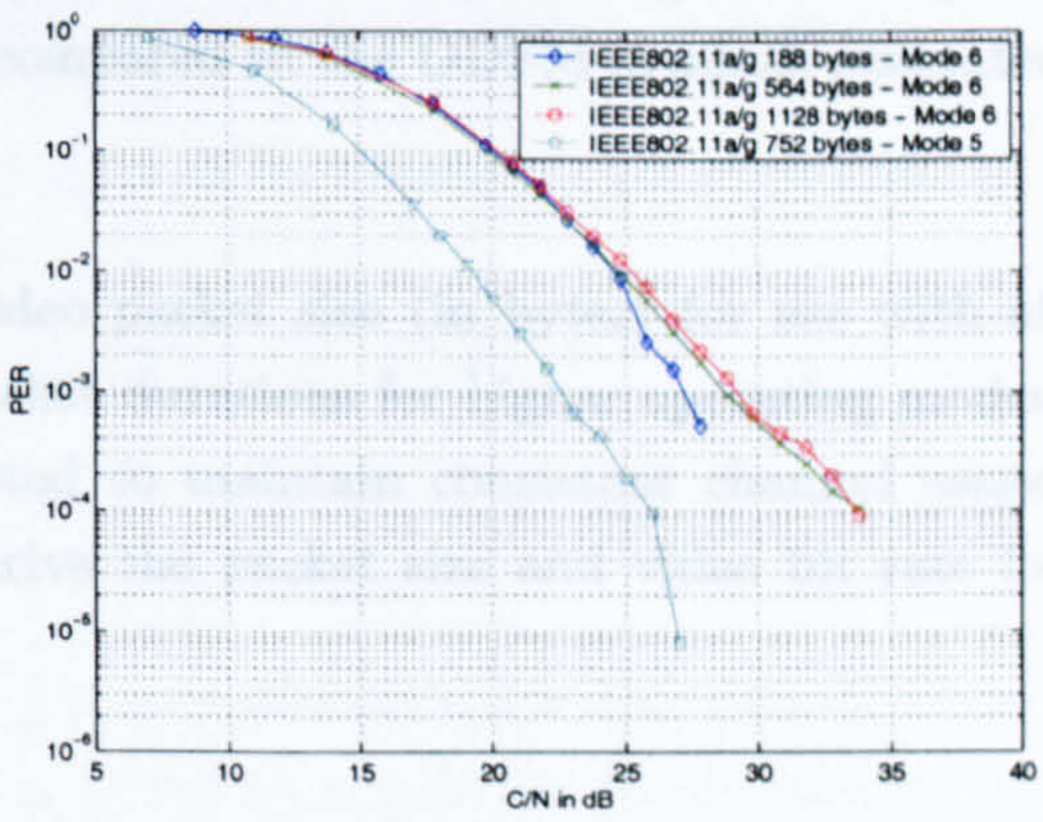
7.3.3 System Description

7.3.3.1 Basis and Assumptions

At a given bit rate, the received video quality highly depends on the PER. As explained in chapter 2, the PER performance of the IEEE 802.11a/g is PHY packet length dependent. For a given mode, a larger packet is more likely to be corrupted. Thus, one way of modifying the perceptual quality is to vary the packet length at the PHY layer at a given operating mode. However, as shown in figure 7.4, changing the mode instead of the PHY packet size has a more dramatic effect on the PER performance at a given C/N level. For e.g., for a C/N of 5dB, mode 1 with a PHY packet size of 188 bytes shows a PER of 2.18×10^{-1}, and a PHY packet size of 1128 bytes gives a PER of 2.91×10^{-1}. Alternatively, a PHY packet size of 188 with mode 3 gives a PER of 5.00×10^{-1}. This is summarised in table 7.3. Therefore, to vary the PER, it is proposed to switch operating modes rather than vary packet lengths. It should be noted however that fine PER control is possible using packet length adaptation within a mode (although this is not considered further in this thesis).



(a) Mode 1 and Mode 3



(b) Mode 5 and Mode 6

Figure 7.4: Influence of Packet Size and Mode on PER performance

The main idea of this investigation is that higher modes (with higher nominal bit rates at the PHY layer) can transmit higher video bit rates and can therefore support better video quality if the PER is sufficiently low. The key point is that each mode will carry one video bit rate offering a scalable transmission depending on the mode chosen.

Table 7.3: Influence of Packet Size and Mode on PER performance

	PER at C/N = 5dB	*PER at C/N = 10dB*
Mode 1 - 188 bytes	2.18×10^{-1}	1.52×10^{-2}
Mode 1 - 564 bytes	2.71×10^{-1}	2.52×10^{-2}
Mode 1 - 1128 bytes	2.91×10^{-1}	3.02×10^{-2}
Mode 3 - 188 bytes	5.00×10^{-1}	7.98×10^{-2}

Whenever the MAC layer adapts its link speed, the application layer also adapts its encoding video bit rate (via a rate control algorithm). For the simulations reported in the following sections, two main assumptions apply:

- The ratios between the bit rates carried on each mode follow the ratios of the nominal bit rates available at the PHY layer for each mode as shown in table 7.4.

- In order to have a fair usage of the channel resources, video packets generate similar numbers of OFDM symbols at the PHY layer. In this way, transmitted packets on each mode have the same use of the radio channel. This ensures that the channel occupancy remains constant irrespective of mode. The ratios between the MAC payload lengths for each mode also respect the ratios of the nominal bit rates available at the PHY layer [165]. This would allow to reduce the timing and delay problems highlighted in the DCF and PCF mode of the IEEE 802.11 MAC (see chapter 3), with unknown time duration. Moreover, this constraint also provides a similar level of error resilience (in terms of slices per frame) at the video codec. Figure 7.5 shows the average number of slices per frame for the sets of video sequences as defined in table 7.10 of section 7.4.1. It can be seen that the average number of slices per frame is consistent from one mode to another for each set. Note that the number of OFDM symbols generated by the RTP/UDP/IP headers are marginal compared to the ODFM symbols generated by the video packets.

It is also possible to assume a fixed video packet size (in bytes) for use with all modes, and this would result in shorter packet durations for higher operating modes. In this thesis, the first solution was adopted to maintain consistent channel usage. Table 7.4 details the equations used to derive the packet size and video bit rate for each mode.

7.3.3.2 Empirical Study

To determine the chosen mode, the following procedure was applied:

- For each mode, the PSNR curve of the received sequence is calculated over a large range of C/N levels. The PSNR was computed as explained in section 6.5 of chapter 6.

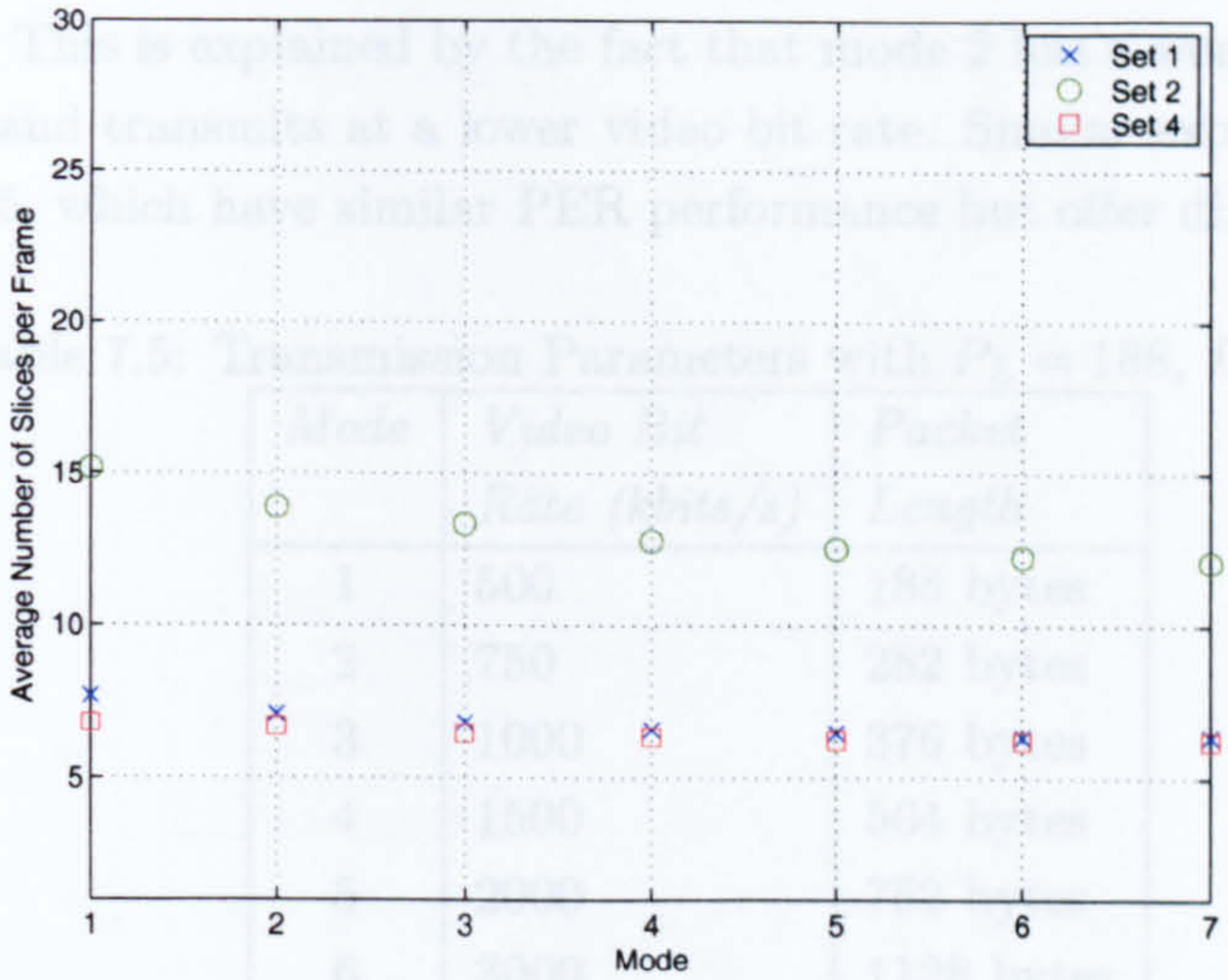


Figure 7.5: Average Number of Slices per Frame

Table 7.4: Packet Length (P_L) and Bit rate (BR) derivation

Mode	*Packet Length*	*Video Bit Rate*
1	$P_{L1} = P_L$	$BR_1 = BR$
2	$P_{L2} = \frac{3}{2} \times P_L$	$BR_2 = \frac{3}{2} \times BR$
3	$P_{L3} = 2 \times P_L$	$BR_3 = 2 \times BR$
4	$P_{L4} = 3 \times P_L$	$BR_4 = 3 \times BR$
5	$P_{L5} = 4 \times P_L$	$BR_5 = 4 \times BR$
6	$P_{L6} = 6 \times P_L$	$BR_6 = 6 \times BR$
7	$P_{L7} = 9 \times P_L$	$BR_7 = 9 \times BR$

- For each power level C/N, the mode that optimises the PSNR is chosen:

$$Mode_{VQ}(C/N) = m \in \{M\} \ with \ PSNR(m, C/N) = max_{i \in \{M\}} \{PSNR(i, C/N)\} \tag{7.3}$$

with $M = \{1, 2, 3, 4, 5, 6, 7\}$ and $Mode_{VQ}$ being the wining mode.

- The crossing points of the PSNR curves define the switching points in term of the C/N at which the system operating mode changes.

Figure 7.6 shows the PSNR curves of the *foreman* sequence, for different C/N levels, with BR = 500 kbits/s and PL = 188 bytes, and with one MAC retransmission (one ARQ). The corresponding bit rates and packet lengths for each mode are shown in table 7.5. As C/N increases, changing to higher modes with a higher bit rate provides a better PSNR. For example, from figure 7.6, a higher PSNR is obtained with mode 5 at a C/N of 20dB, than with mode 3. Mode 5 operates with some errors, whereas mode 3 is able to operate error-free as shown in figure 7.7, but delivers a lower PSNR. A natural and empirical switching point would therefore be based on PSNR; effectively selecting the mode with the highest PSNR at any time and for any C/N level. Note that mode 2 (BPSK 3/4 rate) and mode 4 (QPSK 3/4 rate) will never be chosen for

transmission. This is explained by the fact that mode 2 has a worse PER performance than mode 3 and transmits at a lower video bit rate. Similar explanations apply with modes 4 and 5, which have similar PER performance but offer differing video rates.

Table 7.5: Transmission Parameters with $P_L = 188$, $BR = 500$

Mode	*Video Bit Rate (kbits/s)*	*Packet Length*
1	500	188 bytes
2	750	282 bytes
3	1000	376 bytes
4	1500	564 bytes
5	2000	752 bytes
6	3000	1128 bytes
7	4500	1692 bytes

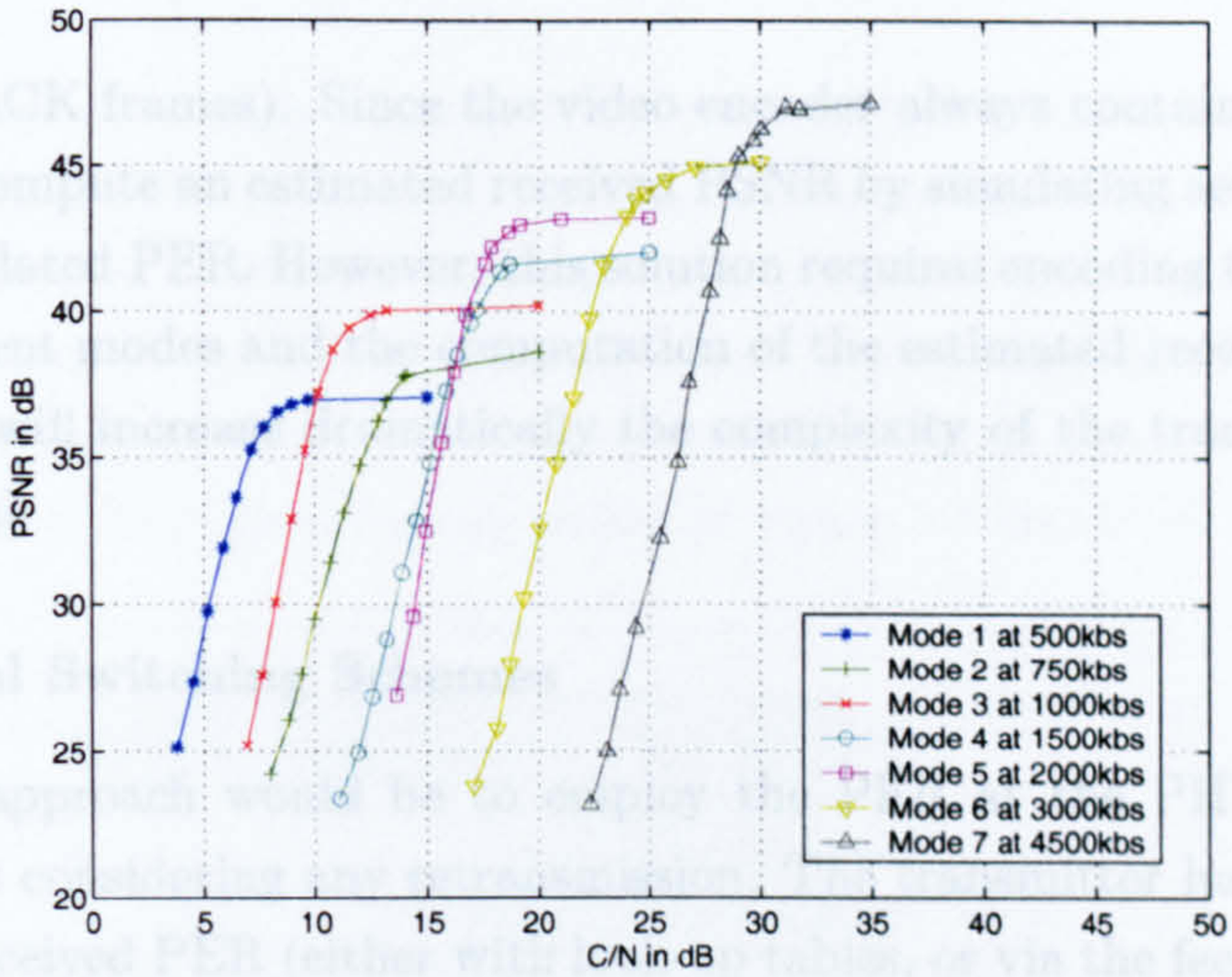


Figure 7.6: Video Quality based algorithm - One ARQ

It should be noted that switching on PSNR is not a practical method for the following two reasons:

- The video quality is only perceived at the receiver. The PSNR calculation requires the original sequence in order to compute the *Mean Square Error* (MSE). In a realistic transmission, the receiver does not have access to the original sequence. It is therefore not possible for the receiver to compute the received PSNR. Also, the transmitter does not have knowledge of the received quality.

- The generation of curves similar to figure 7.6 as look-up tables at the transmitter is not possible since these curves depend on the content, on the bit rate, on the packet length, and on the concealment techniques employed.

The transmitter can have knowledge of the PER either by using the feedback channel, or by using look-up tables, or by simply checking the number of acknowledged

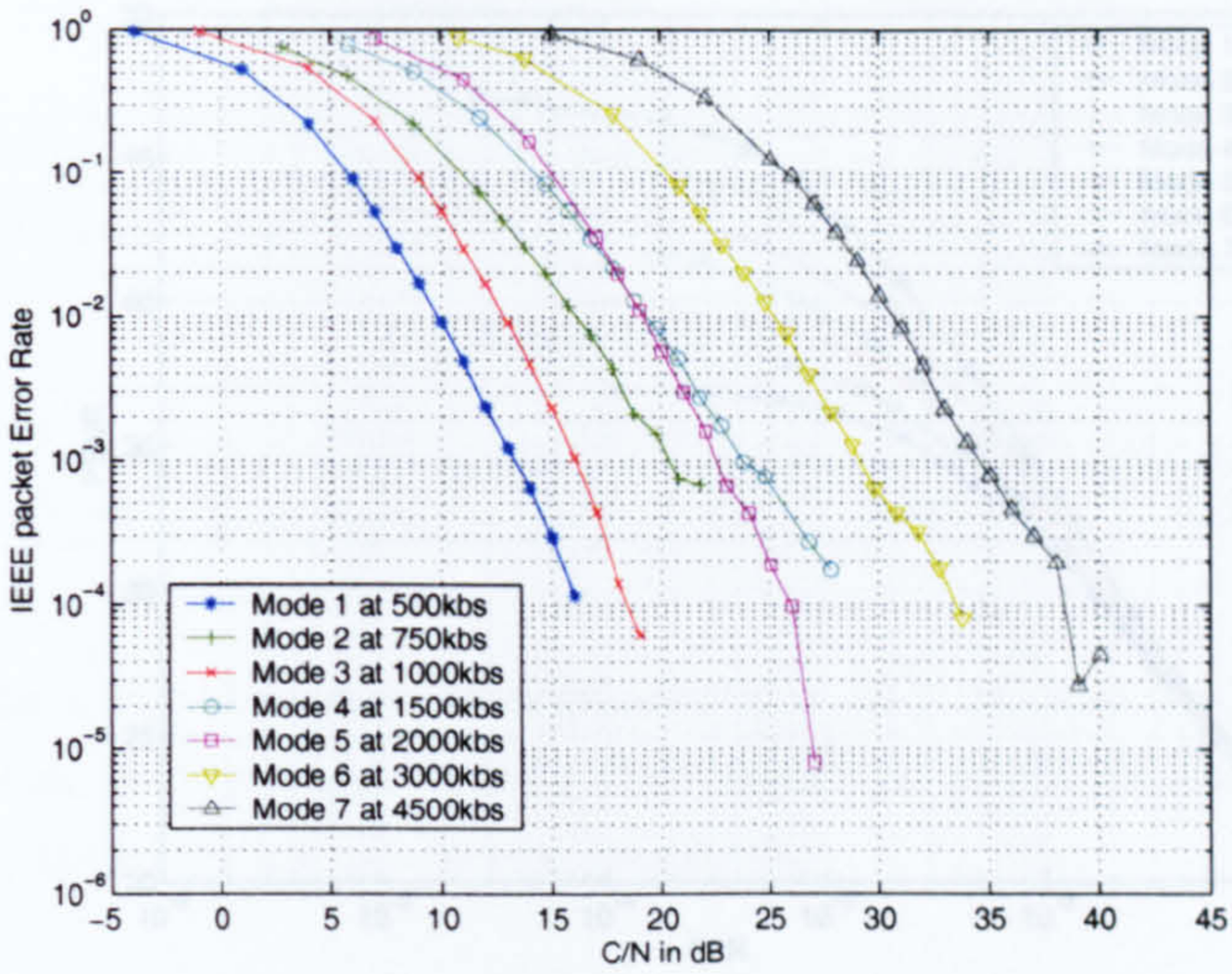


Figure 7.7: PHY PER performances for Foreman - Set 2

packets (received ACK frames). Since the video encoder always contains a decoder, it is also possible to compute an estimated received PSNR by simulating several transmissions with the calculated PER. However, this solution requires encoding two other video sequences on adjacent modes and the computation of the estimated received PSNR for both modes. This will increase dramatically the complexity of the transmitter and is clearly not practical.

7.3.3.3 Practical Switching Schemes

A more practical approach would be to employ the PER at the PHY layer as the metric, i.e. without considering any retransmission. The transmitter has knowledge of the C/N and the received PER (either with look-up tables, or via the feedback channel, or from the number of ACKs received, as explained in section 7.3.3.2). The PSNR and the PER are closely linked as shown in figure 7.8. Note that each mode has different packet lengths and a longer packet is more likely to be corrupted than a smaller packet. It should be noted that the curves of figure 7.8 do not relate to the same packet size.

The following procedure applies for the choice of the operating mode:

- $PER(m, C/N) \leq PER_u(m) : \ Up - scale \ if \ m \neq 7$
- $PER_d(m) \leq PER(m, C/N) : \ Down - scale \ if \ m \neq 1$
- $PER_u(m) \leq PER(m, C/N) \ and \ PER(m, C/N) \leq PER_d(m) : \ stay \ in \ the \ current \ mode$

where $PER(m, C/N)$ defines the PHY PER with the current operating mode m at a given C/N level, and where $PER_u(m)$ and $PER_d(m)$ define the PER thresholds to up-scale to a higher mode and down-scale to a lower mode respectively. If the current PHY $PER(m, C/N)$ is lower than a first predefined threshold $PER_u(m)$, then the

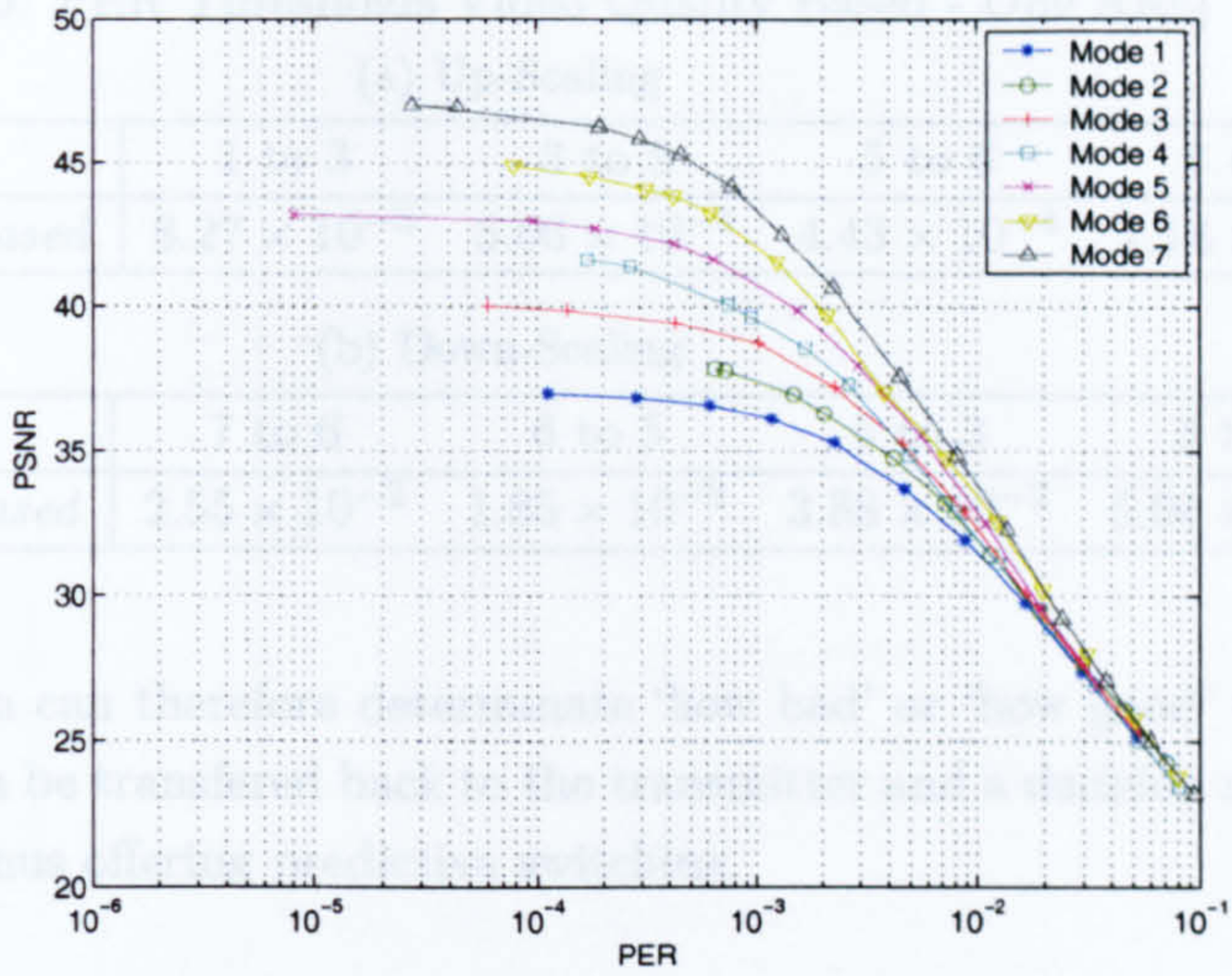


Figure 7.8: PSNR versus PER - Foreman - Set 2 - No ARQ

transmitter switches to a higher mode. If the current PHY PER is higher than a second predefined threshold $PER_d(m)$, then the transmitter switches to a lower mode. If the current PHY $PER(m, C/N)$ lies between these two thresholds, then the transmitter stays with the same mode. Figure 7.9 and table 7.6 show the different up-scaling and down-scaling PER thresholds for the above example. These thresholds were obtained by converting the C/N of the crossing points of figure 7.6 into PER using figure 7.7.

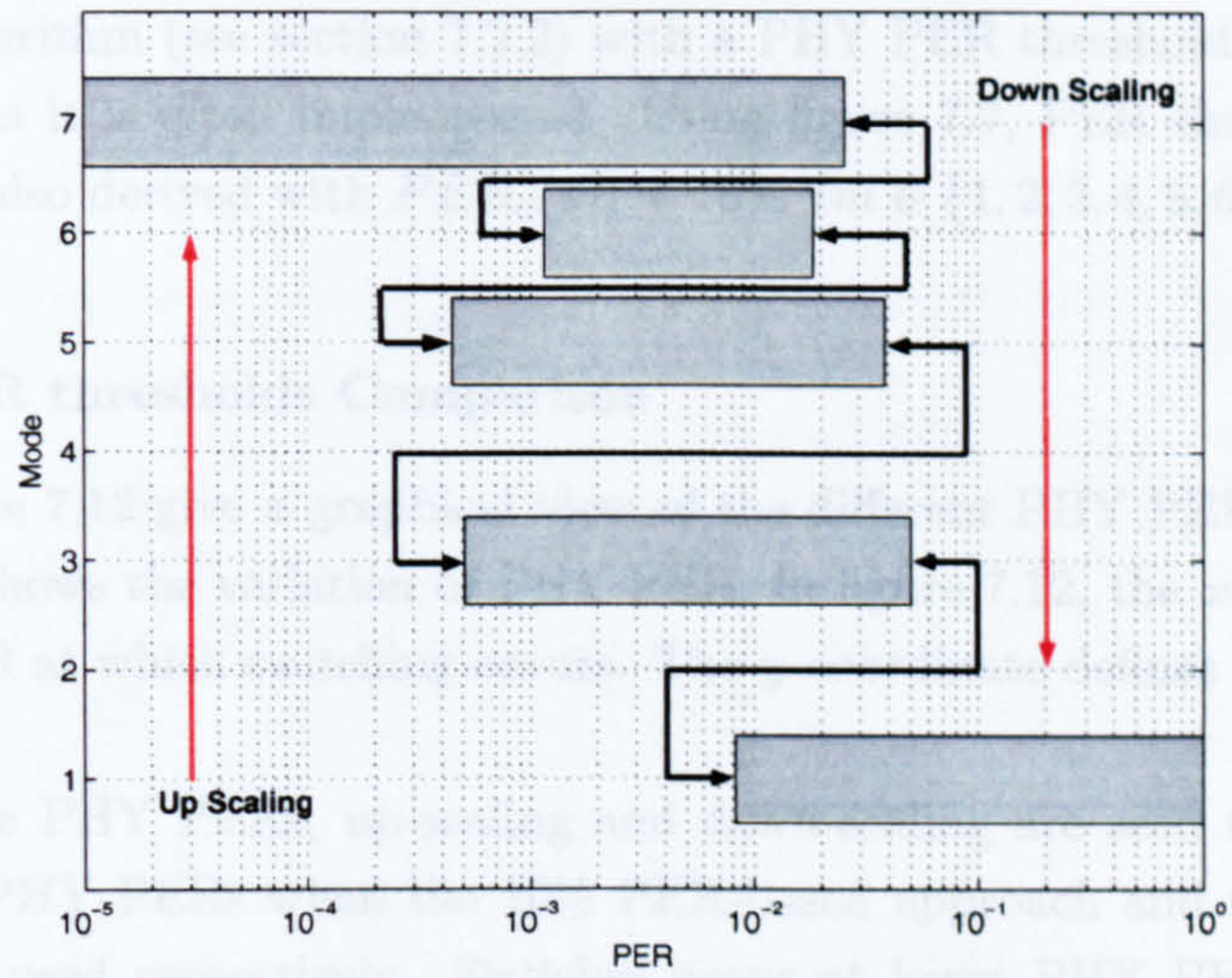


Figure 7.9: Switching Points - Video Quality based - One ARQ

Another possible switching metric for a link adaptation scheme would be to use the *Error Vector Magnitude* (EVM), indicating the error margin of the received signal. PER is based on a symbol error rate (on the modulation constellation). It therefore defines the packet as correct or corrupted. The EVM gives soft information on the value of the received symbol (i.e. prior to symbol detection) compared to the correct

Table 7.6: PER Thresholds Video Quality Based - One ARQ

(a) Up-Scaling

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
Video Quality based	8.27×10^{-3}	5.06×10^{-4}	4.43×10^{-4}	1.14×10^{-3}

(b) Down-Scaling

From Mode	7 to 6	6 to 5	5 to 3	3 to 1
Video Quality based	2.55×10^{-2}	1.85×10^{-2}	3.88×10^{-2}	5.04×10^{-2}

one. The information can therefore determinate 'how bad' or 'how good' the channel currently is. This can be transfered back to the transmitter and a decision made before the PER degrades, thus offering predictive switching.

7.3.4 A Comparison of Algorithms

In this section, the presented algorithm is compared with two existing link adaptation algorithms implemented in IEEE 802.11 cards:

- *SNR (C/N) and Throughput-based algorithm* (see section 7.2.4): This algorithm optimises the error-free throughput (as expressed in equation 7.1) and makes an extensive use of MAC layer ARQ. Figure 7.10 shows the throughput-based algorithm applied with the PHY packet size being derived with PL, as defined in table 7.5. As explained, PER thresholds can be derived by converting the C/N switching points into PER using figure 7.7.

- PER-based algorithm (see section 7.2.2) with a PHY PER threshold of 10% for down-scaling, as it is often implemented. Using figure 7.7, PER thresholds for up-scaling are also derived with $PER_d(m) = 10\% \; \forall m \in \{1,2,3,4,5,6,7\}$

7.3.4.1 PHY PER thresholds Comparison

Figure 7.11 and figure 7.12 give a graphical view of the different PHY PER threshold points. Figure 7.11 shows the variation of PHY PER. In figure 7.12, the x-coordinate defines the PHY PER at which switching occurs. The y-coordinate defines the current mode.

By comparing the PHY PERs, up-scaling and down-scaling are seen to occur at high and very high PHY PERs when the 10% PER-based approach and throughput based algorithm are used respectively. Switches occur at lower PHY PERs for the video quality based algorithm as shown in figure 7.12. These figures show that a video application, which in the case is constrained to a single retransmission (one MAC ARQ), would up-scale from mode 3 to mode 5 when the PHY PER on mode 3 reaches 4.5×10^{-2} and 2.23×10^{-3} if the system uses the throughput-based and 10% PER-based link adaptation schemes respectively. The empirical study conducted shows that the switching should occur when the PHY PER on mode 3 reaches 5×10^{-4}. On the

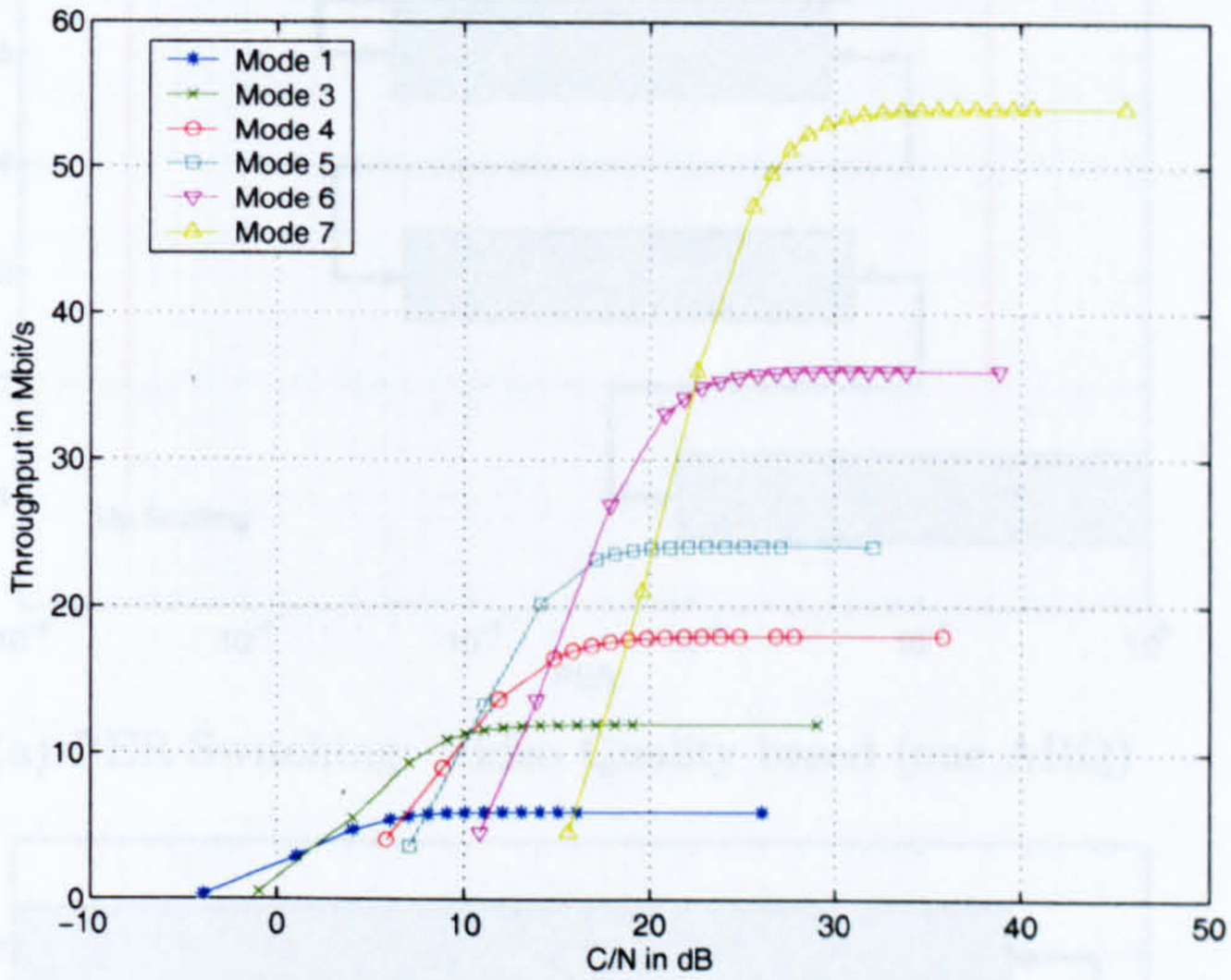
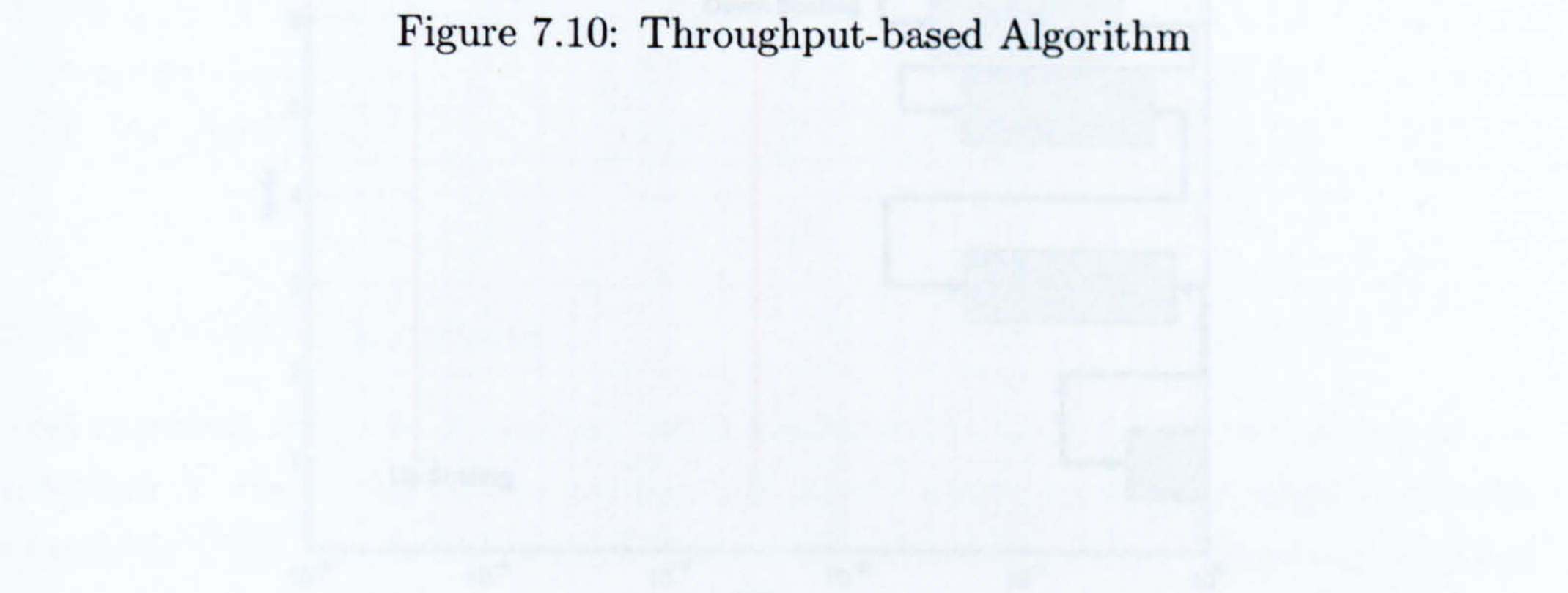


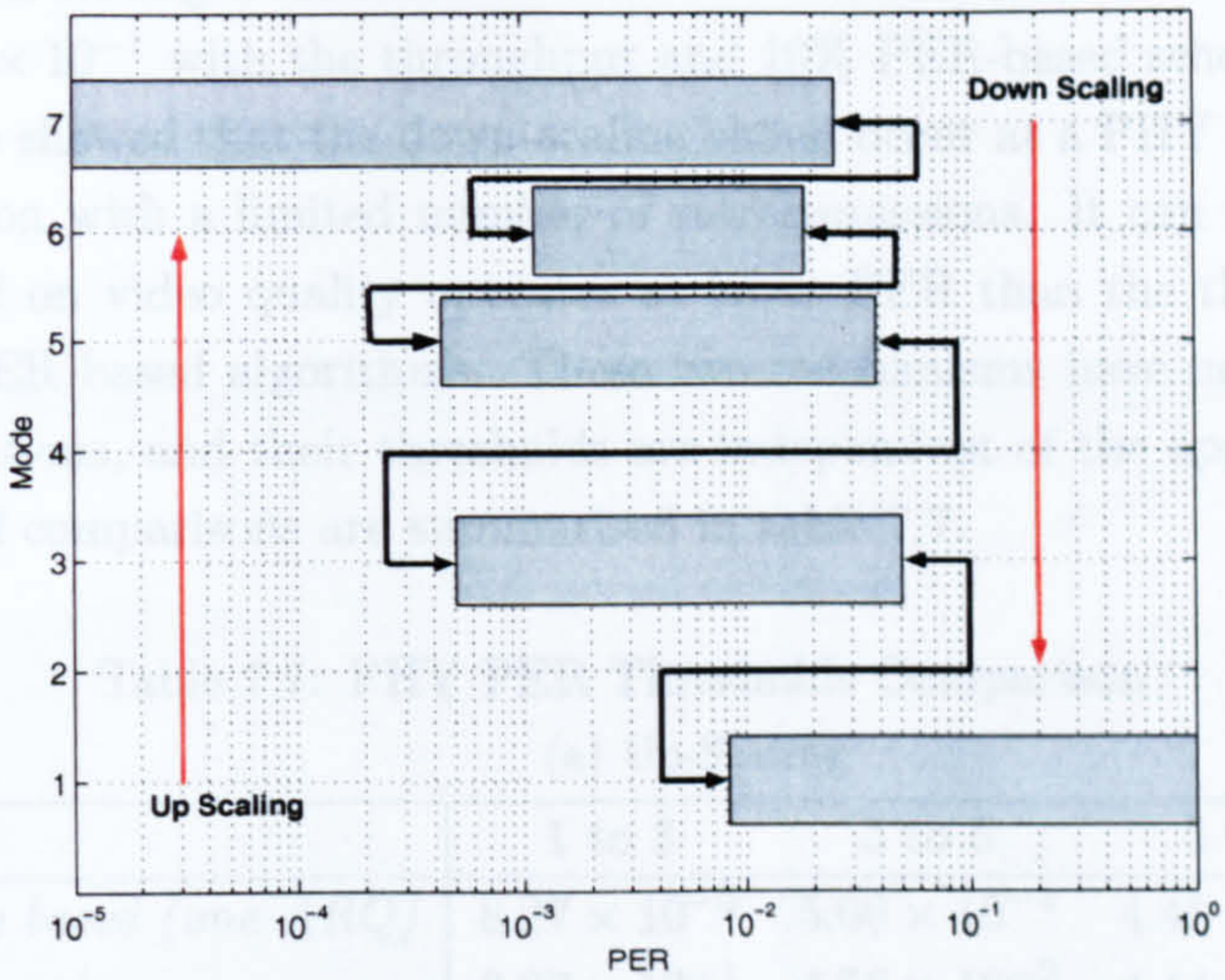
Figure 7.10: Throughput-based Algorithm



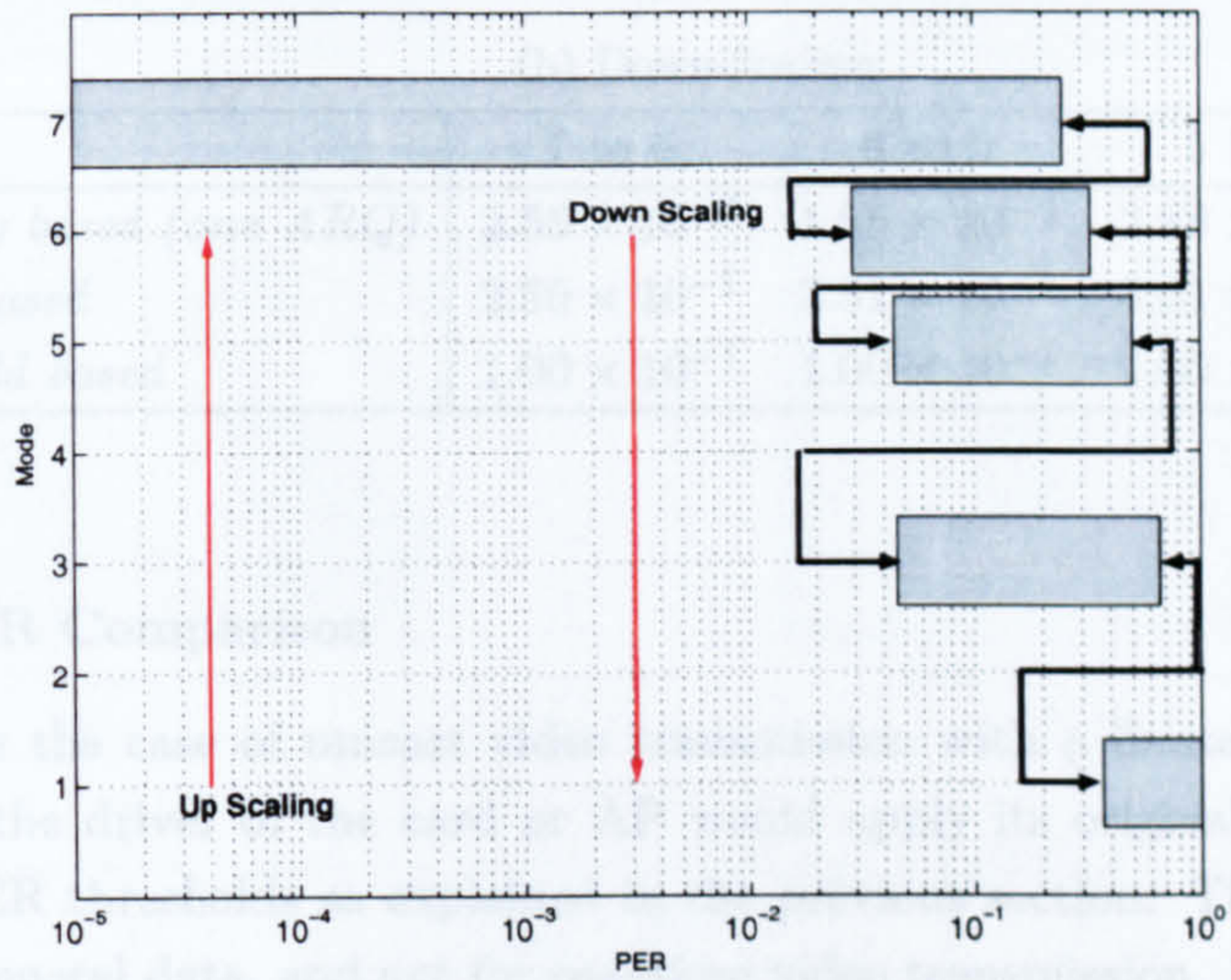
(a) Video Quality - Throughput based Comparison

(b) Video Quality - 10% PER based Comparison

Figure 7.11: Link Adaptation PER Comparison



(a) PER Switching: Video Quality based (one ARQ)



(b) PER Switching: Throughput based

(c) PER Switching: 10% PHY PER based

Figure 7.12: Switching Points Comparison - Foreman

other hand, down-scaling from mode 5 to mode 3 occurs when the PHY PER reaches 5×10^{-1} and 1×10^{-1} with the throughput and 10% PER-based scheme respectively. The studies also showed that the down-scaling shoud occur at a PHY PER of 4×10^{-2} for an application with a limited number of retransmissions. It can be seen that the algorithm based on video quality operates at lower PER than the throughput based and the 10% PER based algorithms. These two mechanisms have not been designed for such applications, and their thresholds are independent of the application or data type. Threshold comparisons are summarised in table 7.7.

Table 7.7: PHY PER Thresholds Comparison

(a) Up-Scaling

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
Video Quality based (one ARQ)	8.27×10^{-3}	5.06×10^{-4}	4.43×10^{-4}	1.14×10^{-3}
Throughput based	3.67×10^{-1}	4.56×10^{-2}	4.44×10^{-2}	2.94×10^{-2}
10% PER based	1.97×10^{-2}	2.23×10^{-3}	5.01×10^{-3}	4.67×10^{-3}

(b) Down-Scaling

From Mode	7 to 6	6 to 5	5 to 3	3 to 1
Video Quality based (one ARQ)	2.55×10^{-2}	1.85×10^{-2}	3.88×10^{-2}	5.04×10^{-2}
Throughput based	2.50×10^{-1}	3.31×10^{-1}	5.01×10^{-1}	6.64×10^{-1}
10% Threshold based	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}

7.3.4.2 PSNR Comparison

If one considers the case of unicast video transmission with a limited number of retransmissions, the driver of the card or AP would apply its original algorithm with its pre-built PER thresholds as explained in the previous section. These schemes are developed for general data, and not for real-time video transmission. Figure 7.13 compares the PSNR of the received sequence for the throughput based algorithm with the empirical study conducted optimising the video quality with one retransmission permitted, along with picture snapshots. For both schemes, for a high C/N, mode 7 is chosen, providing the highest video quality. However, as the receive power decreases and the PHY PER increases, the throughput based algorithm remains with this mode, whereas the proposed algorithm switches earlier (at a lower PER) to a lower mode to maintain the video quality.

Similar comments apply to figure 7.14 comparing the empirical video quality study with the 10% PER-based algorithm. The average received PSNR for the three studies are presented in table 7.8. It shows that a 7dB gain in average PSNR is possible when the PER thresholds are low. Because the PHY thresholds are lower for the 10% PER-based system than for the throughput-based case, the PSNR performance is better. Clearly, the high PHY PER thresholds do not allow good video quality reconstruction at the decoder and a lowering the PER thresholds puts the switching points in the operating range of the video codec for real-time applications that require low latency

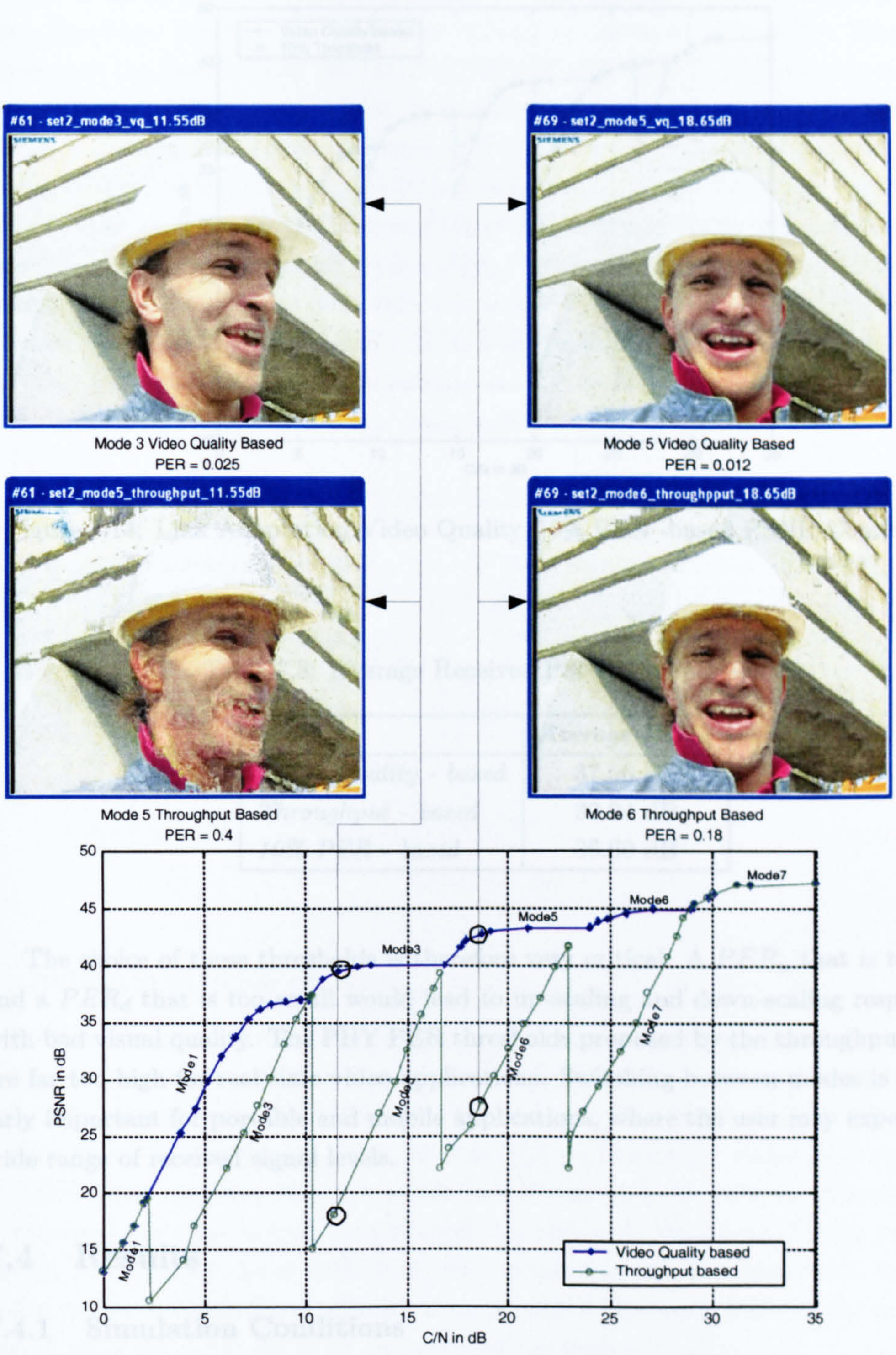


Figure 7.13: Link Adaptation Video Quality/Throughput -based PSNR Comparison

and therefore use a low number of ARQs.

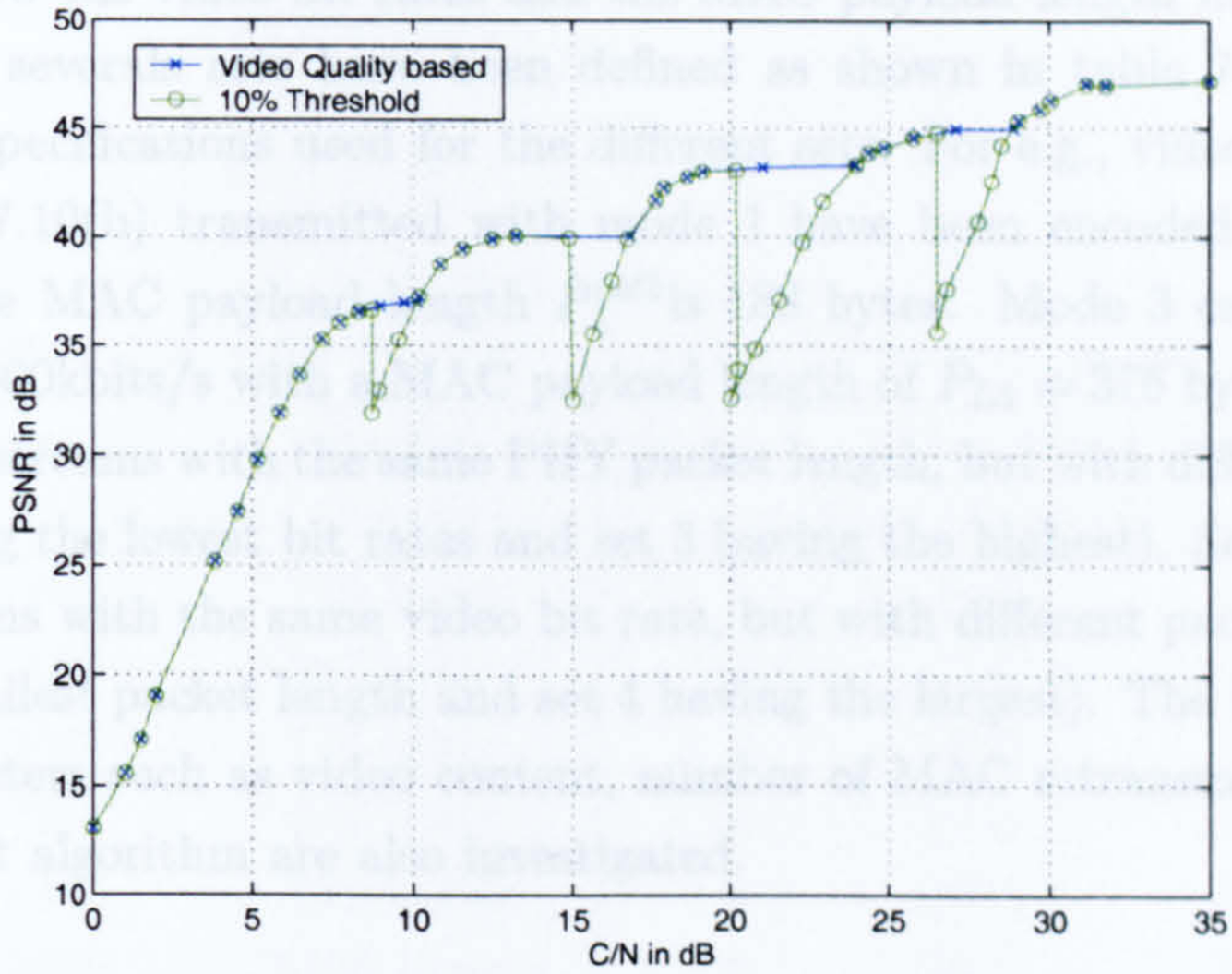


Figure 7.14: Link Adaptation Video Quality/10% PER -based PSNR Comparison

Table 7.8: Average Received PSNR Comparison

Scheme	*Average PSNR*
Video Quality - based	37.56 dB
Throughput - based	30.94 dB
10% PER - based	36.60 dB

The choice of these thresholds is therefore very critical. A PER_u that is too large and a PER_d that is too small would lead to up-scaling and down-scaling respectively with bad visual quality. The PHY PER thresholds proposed by the throughput metric are far too high for real-time video applications. Switching between modes is particularly important for portable and mobile applications, where the user may experience a wide range of received signal levels.

7.4 Results

7.4.1 Simulation Conditions

It was shown in the previous section that the optimum mode switching points that maximise video quality for low delay transmission are different from those that maximise the error-free throughput. In this section, empirical simulation results obtained by modifying parameters such as the content of the video sequences, the video bit rate, the packet length, the number of MAC ARQ and the error resilience technique are presented. Results are obtained with the PSNR obtained by averaging over 100 sequences for each point. The conditions for the simulations were presented in sections

6.5 and 7.3.3. Unless specified, the error concealment algorithm used at the decoder is the *Advanced Error Concealment Algorithm* (AEC) as described in section 4.4.4.

In order to see how the video bit rates and the MAC payload length influence the proposed algorithm, severals sets have been defined as shown in table 7.10. Table 7.9 summarises the specifications used for the different sets. For e.g., video sequences from set 2 in table 7.10(b) transmitted with mode 1 have been encoded with BR_1 = 500kbits/s and the MAC payload length $P_L^{set1}$is 188 bytes. Mode 3 carries video encoded at BR_3 = 1000kbits/s with a MAC payload length of P_{L3} = 376 bytes. Sets 1, 2 and 3 use video bit-streams with the same PHY packet length, but with different video bit rates (set 1 having the lowest bit rates and set 3 having the highest). Sets 2, 4 and 5 use video bit-streams with the same video bit rate, but with different packet lengths (set 5 having the smallest packet length and set 4 having the largest). The influence of various other parameters such as video content, number of MAC retransmissions and the error concealment algorithm are also investigated.

Table 7.9: Packet Length (P_L) and Bit Rate (BR) Specifications

(a) Parameters Allocation

Set	P_L *in bytes*	BR *in kbits/s*
Set 1	188	250
Set 2	188	500
Set 3	188	1000
Set 4	376	500
Set 5	94	500

(b) Parameters Comparison

$$P_L^{set1} = P_L^{set2} = P_L^{set3}$$
$$P_L^{set4} < P_L^{set5} < P_L^{set5}$$
$$BR^{set1} < BR^{set2} < BR^{set3}$$
$$BR^{set2} = BR^{set4} = BR^{set5}$$

7.4.2 Influence of video content

In this section, the effect of the video content on the switching points is studied. Parameters of set 2 have been used to encode the *akiyo* (slow motion), *foreman* (medium motion), *table* (fast motion) and *stefan* (very fast motion). The MAC payload length is the same for all four sequences, giving similar PER performances. *Akiyo* produces higher PSNRs than the three other sequences for a similar video bit rate. In contrast, *stefan* presents very high motion activity, and this results in the lowest PSNRs for a given bit rate (mode). From table 7.11, we can see that the switching levels for up-scaling and down-scaling respectively occur at almost the same PER for all the sequnences using the *Advanced Error Concealment*(see section 7.4.6).

Figures 7.15(a) and 7.15(b) plot the up-scaling and down-scaling switching points respectively for the four sequences along with the corresponding values for the throughput based algorithm. On these figures, the y-coordinate corresponds to the mode after switching. These figures show that switching (up-scaling and down-scaling) occurs at very similar PER for all the video sequences. It also confirms that the throughput

Table 7.10: Video Sets

(a) Set 1 - $P_L^{set1} = 188$, $BR^{set1} = 250$

Mode	*Video Bit Rate (kbits/s)*	*Packet Length*
1	250	188 bytes
2	375	282 bytes
3	500	376 bytes
4	750	564 bytes
5	1000	752 bytes
6	1500	1128 bytes
7	2250	1692 bytes

(b) Set 2 - $P_L^{set2} = 188$, $BR^{set2} = 500$

Mode	*Video Bit Rate (kbits/s)*	*Packet Length*
1	500	188 bytes
2	750	282 bytes
3	1000	376 bytes
4	1500	564 bytes
5	2000	752 bytes
6	3000	1128 bytes
7	4500	1692 bytes

(c) Set 3 - $P_L^{set3} = 188$, $BR^{set3} = 1000$

Mode	*Video Bit Rate (kbits/s)*	*Packet Length*
1	1000	188 bytes
2	1500	282 bytes
3	2000	376 bytes
4	3000	564 bytes
5	4000	752 bytes
6	6000	1128 bytes
7	9000	1692 bytes

(d) Set 4 - $P_L^{set4} = 376$, $BR^{set4} = 500$

Mode	*Video Bit Rate (kbits/s)*	*Packet Length*
1	500	376 bytes
2	750	564 bytes
3	1000	752 bytes
4	1500	1128 bytes
5	2000	1508 bytes
6	3000	2256 bytes
7	4500	3384 bytes

(e) Set 5 - $P_L^{set5} = 94$, $BR^{set5} = 500$

Mode	*Video Bit Rate (kbits/s)*	*Packet Length*
1	500	94 bytes
2	750	141 bytes
3	1000	188 bytes
4	1500	282 bytes
5	2000	376 bytes
6	3000	564 bytes
7	4500	846 bytes

Table 7.11: Influence of Video Content - PER Thresholds - Set 2 - 1 ARQ

(a) Up-Scaling

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
Akiyo	8.17×10^{-3}	5.29×10^{-4}	4.89×10^{-4}	1.48×10^{-3}
Foreman	8.27×10^{-3}	5.07×10^{-4}	4.43×10^{-4}	1.15×10^{-3}
Table	6.09×10^{-3}	6.01×10^{-4}	3.64×10^{-4}	8.59×10^{-4}
Stefan	1.05×10^{-2}	6.41×10^{-4}	5.98×10^{-4}	1.34×10^{-3}

(b) Down-Scaling

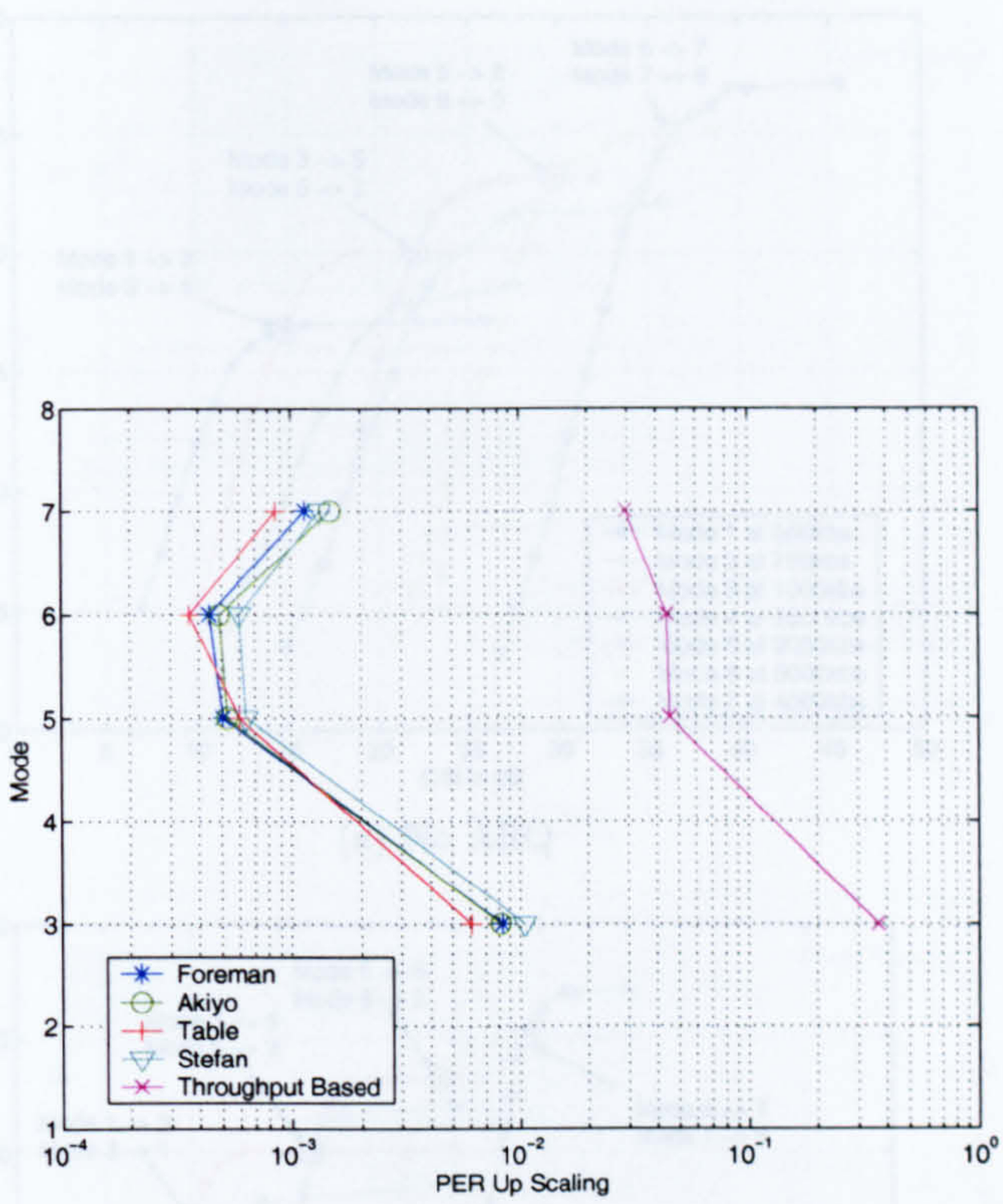
From Mode	7 to 6	6 to 5	5 to 3	3 to1
Akiyo	3.05×10^{-2}	2.13×10^{-2}	3.80×10^{-2}	4.93×10^{-2}
Foreman	2.55×10^{-2}	1.85×10^{-2}	3.88×10^{-2}	5.04×10^{-2}
Table	1.99×10^{-2}	1.63×10^{-2}	4.26×10^{-2}	3.72×10^{-2}
Stefan	2.83×10^{-2}	2.60×10^{-2}	4.43×10^{-2}	6.14×10^{-2}

based algorithm switches at very high PERs. The fact that similar switching points are obtained for different sequences can be explained because even if PSNR curves are different from one sequence to another, switching points are obtained by comparing one sequence at one mode with the same sequence at another mode.

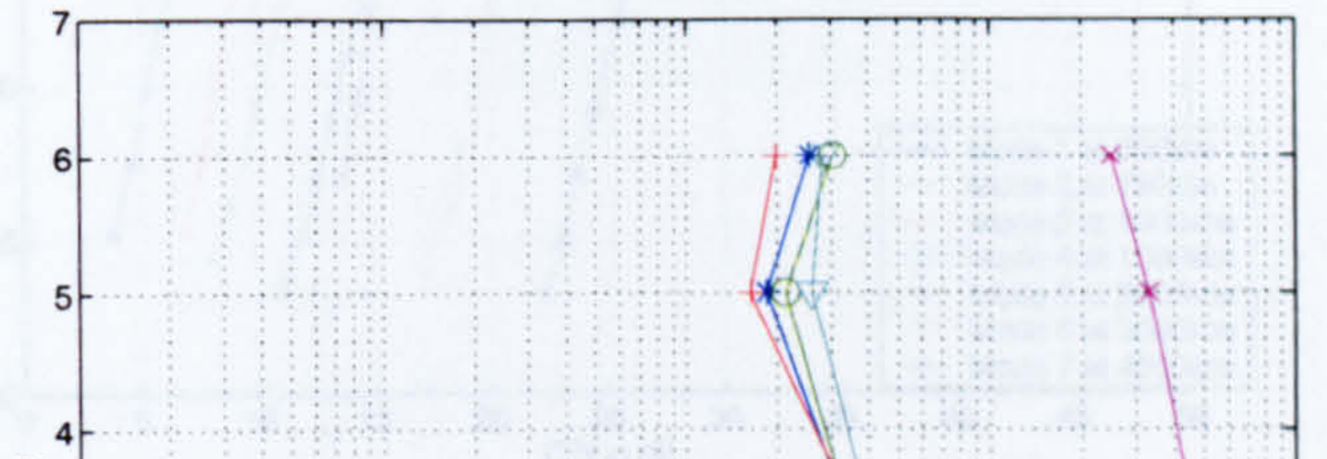
7.4.3 Influence of the number of Layer 2 ARQs (MAC layer)

In this section, the effect of the number of MAC layer ARQs on the mode switching points is studied. Parameters from set 2 have been used for the *foreman* sequence. Figure 7.16 shows the influence of ARQs on the PSNR performance. As the number of ARQs increases, the PSNR curves are shifted left and a given PSNR is achieved at a lower C/N. Switching from one mode to another is therefore influenced by ARQ as shown in figure 7.17.

Note that, in the case of a unicast transmission with no ARQ, apart from switching from mode 1 to 3, an almost error free transmission channel is required in order to switch to higher modes. However, it can be seen that as the number of ARQs increases, switching (up or down) can be performed at higher PERs. This is also shown in figure 7.18 by plotting the up and down-scaling points respectively. As more ARQs are allowed, the switching points move closer to those of the throughput based algorithm. The down-scaling PER threshold from mode 5 to mode 3 is at 1.5×10^{-3} with no ARQ, 4×10^{-2} with one ARQ, and 1×10^{-1} with two ARQs. The number of ARQs that the application can tolerate therefore determines the switching thresholds. Table 7.12 summarises the up and down-scaling PER thresholds for different numbers of ARQ, for *foreman*, set 2. Ultimately, as the number of ARQ increases up to a maximum of 32, the switching points of the proposed algorithm would match the switching points of the throughput-based scheme.

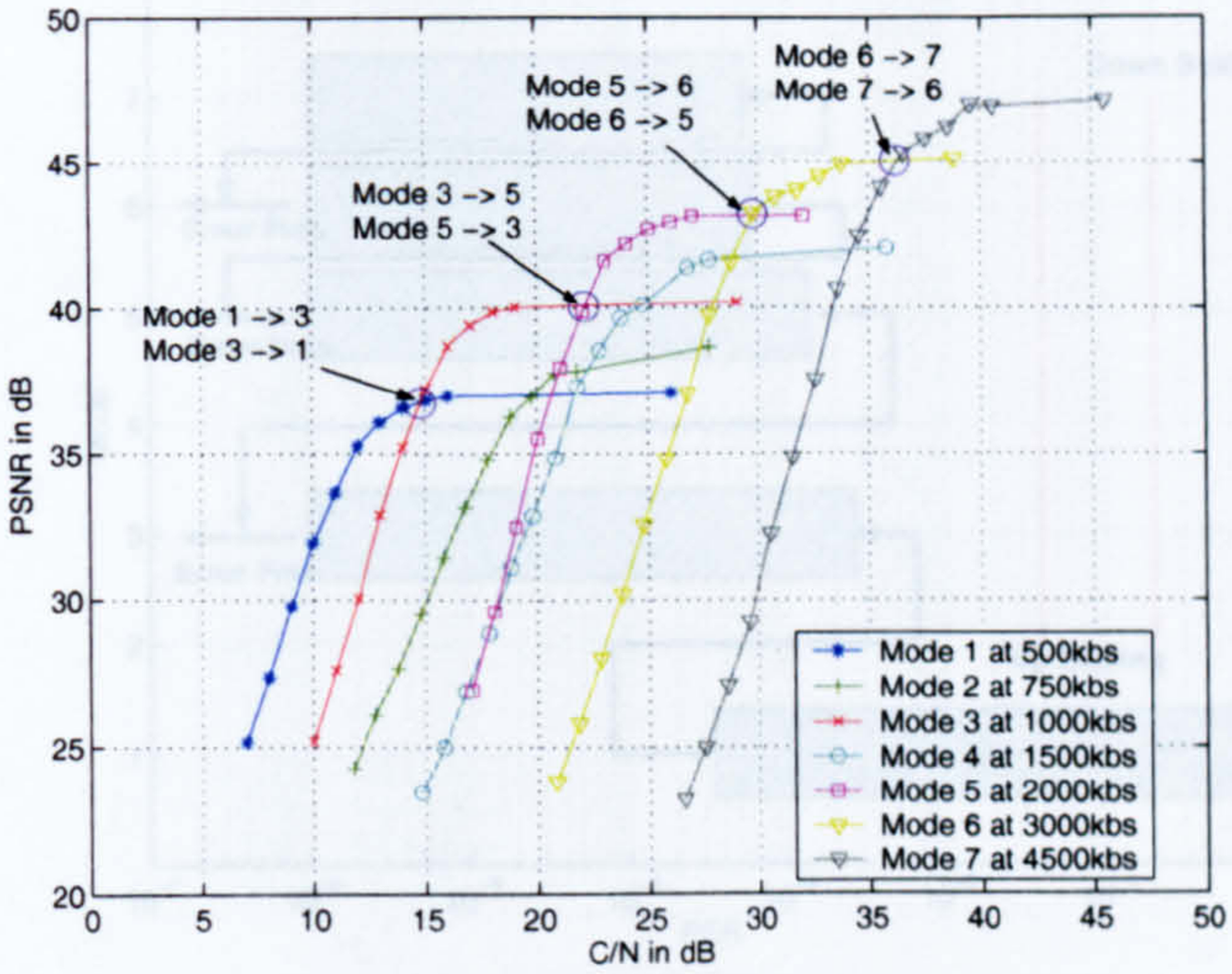


(a) PER switching: Up-scaling

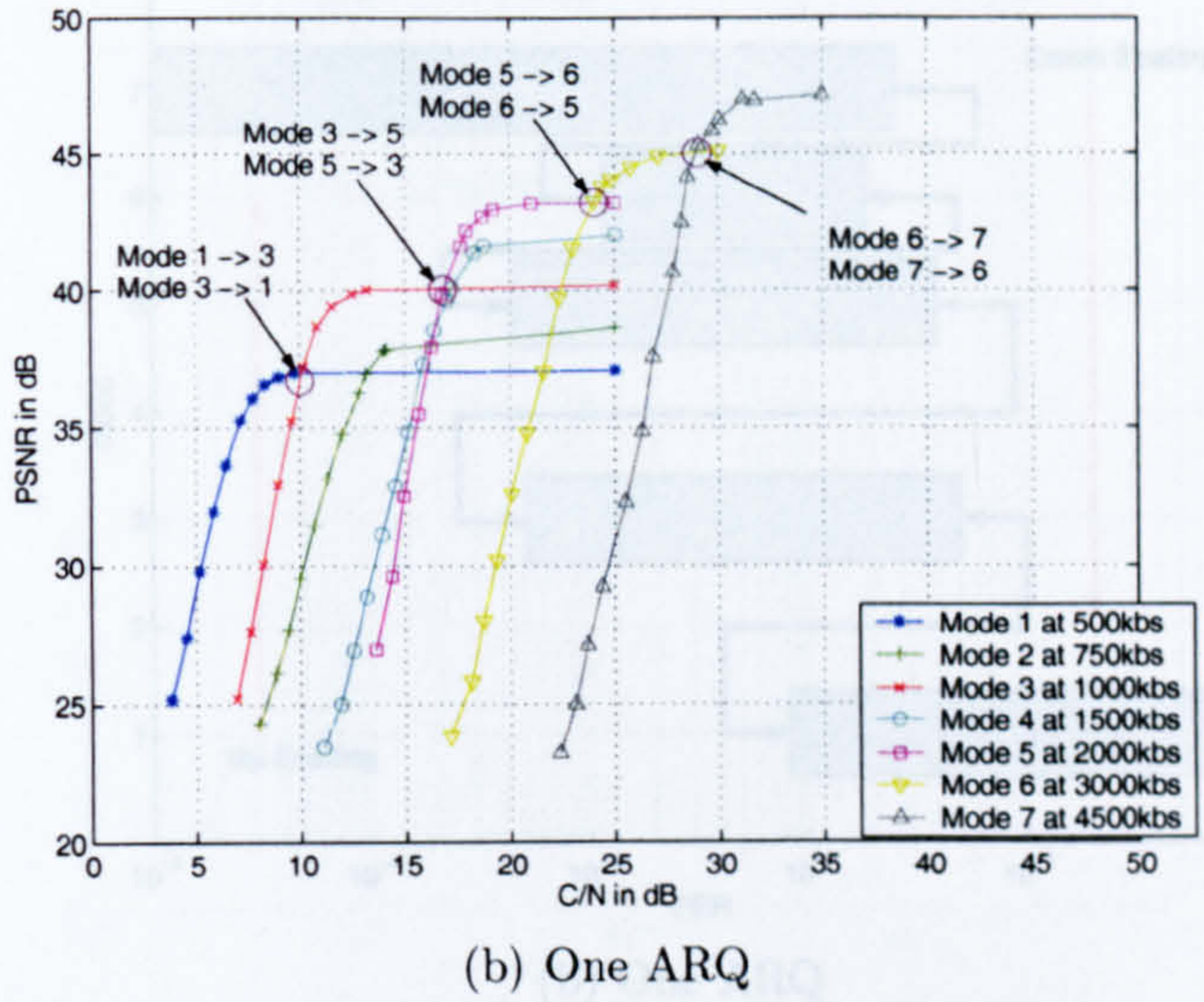


(b) PER switching: Down-scaling

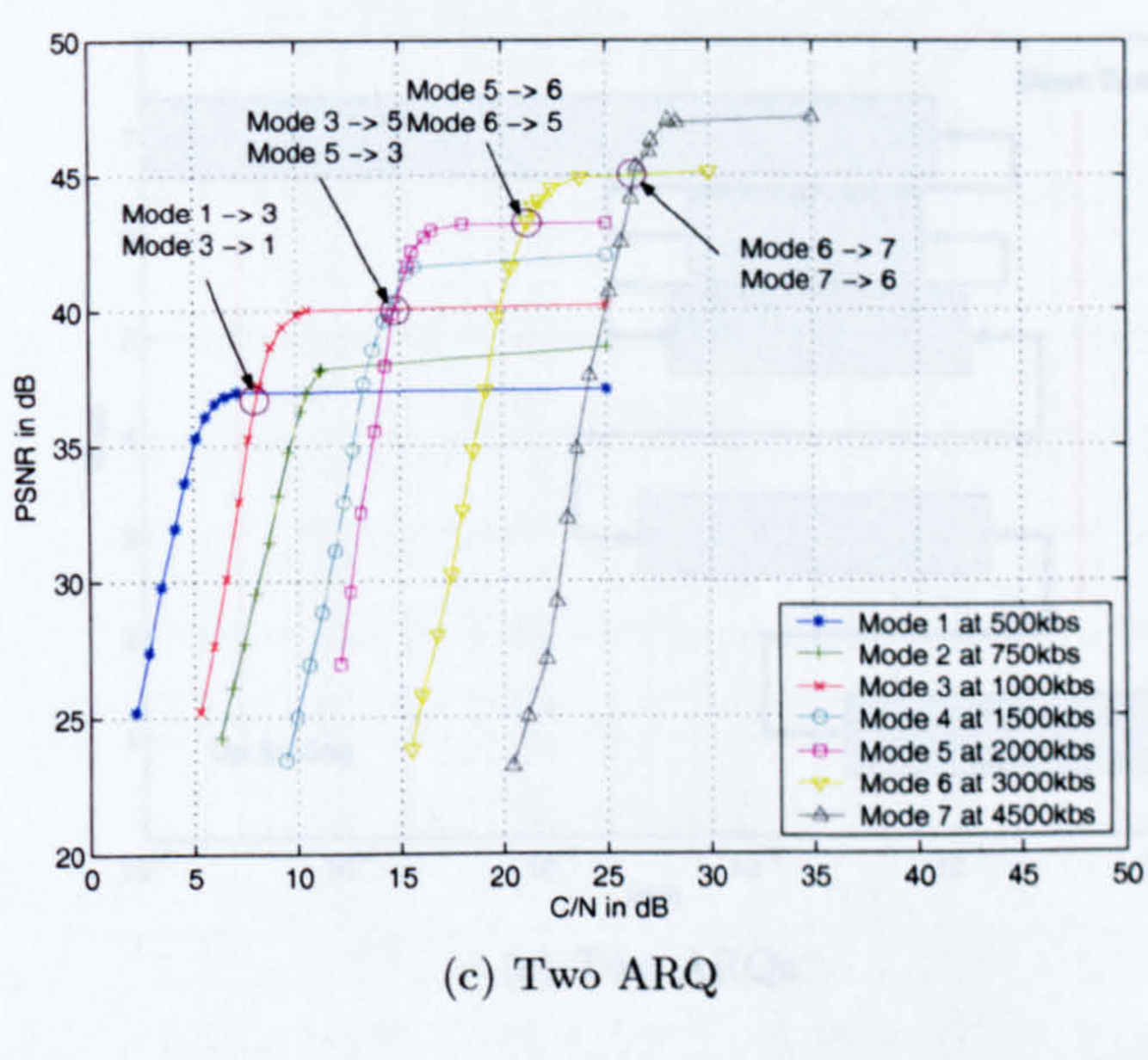
Figure 7.15: Video Content Comparison - PER based switching points - Set 2 - One ARQ



(a) No ARQ

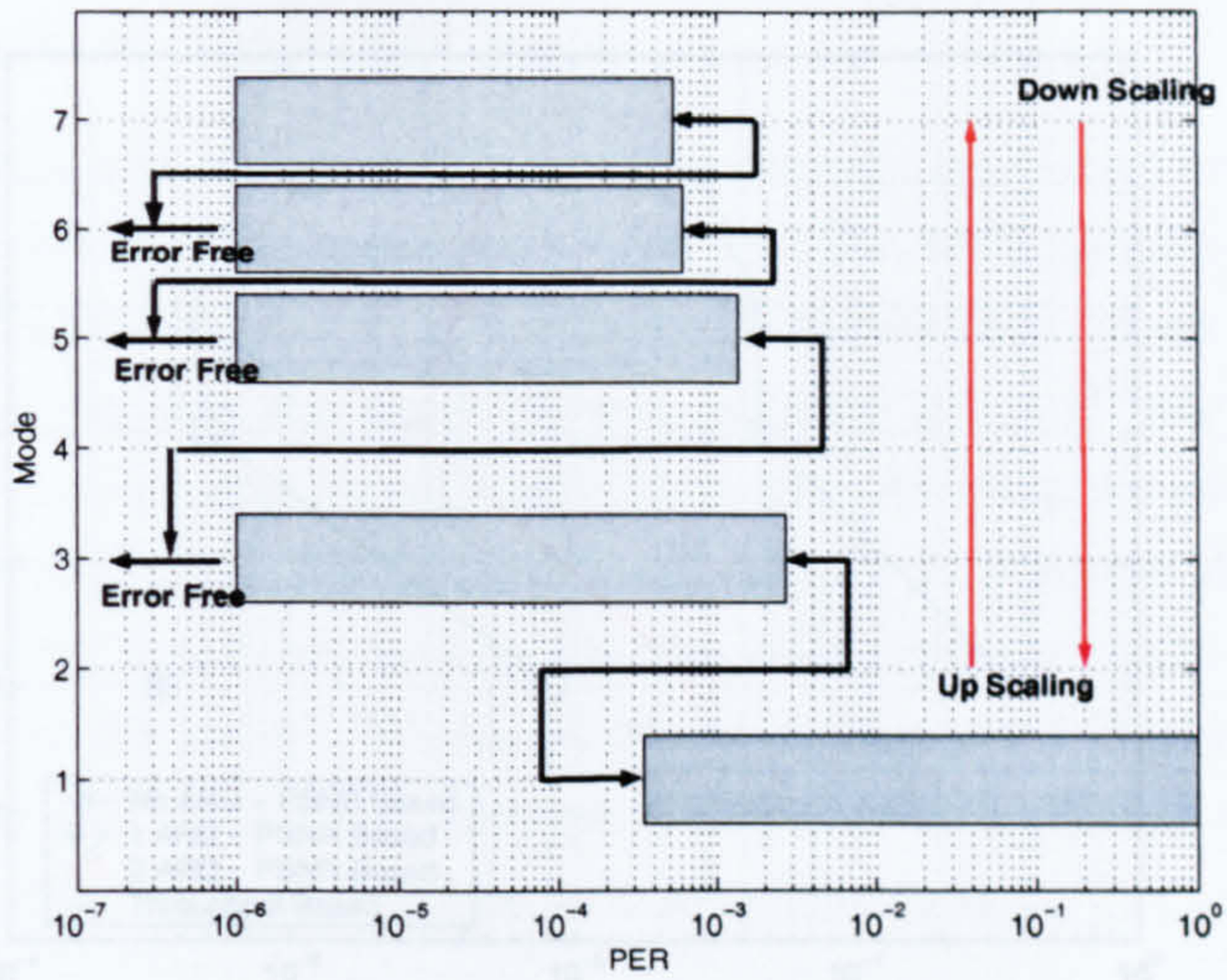


(b) One ARQ

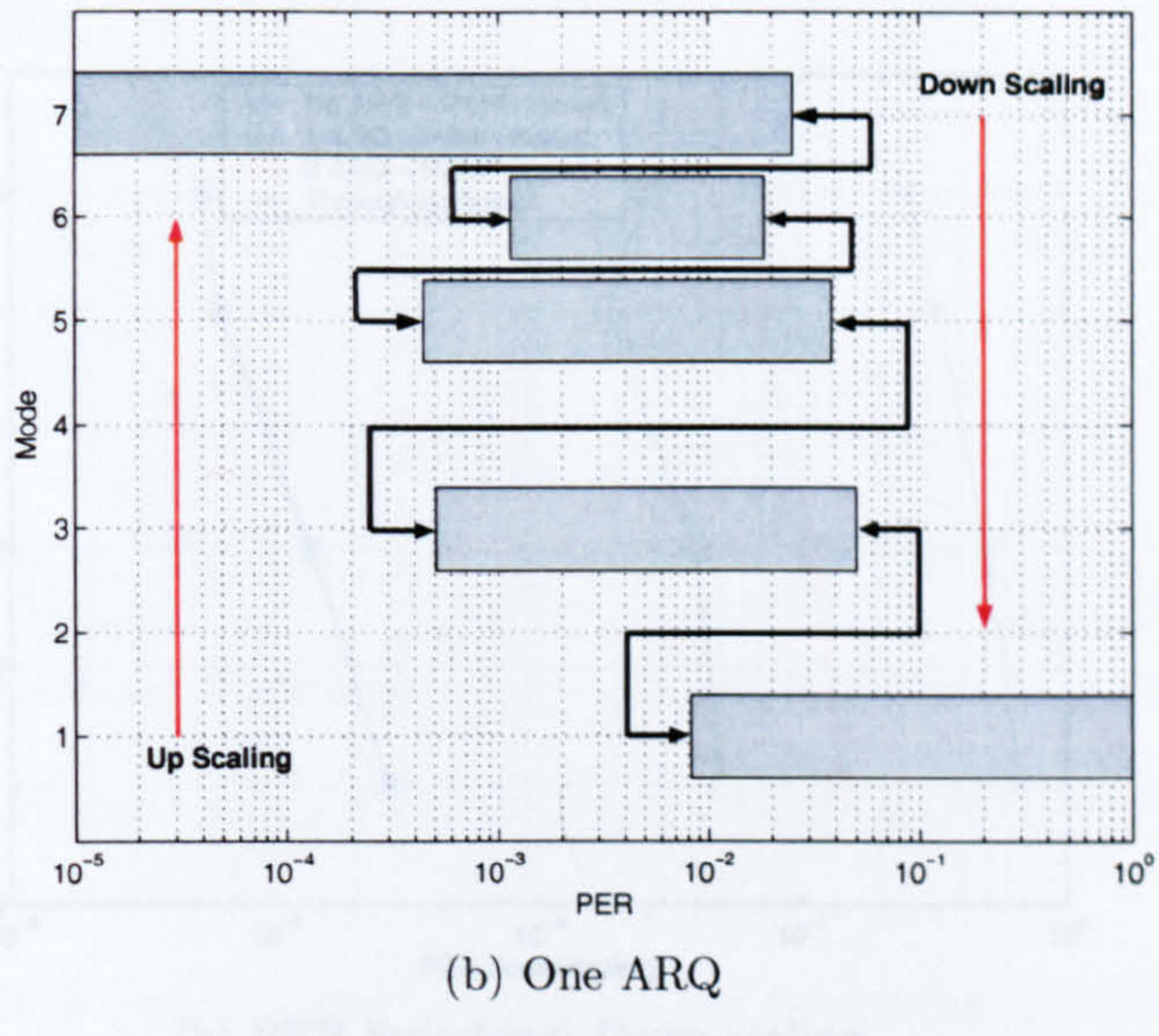


(c) Two ARQ

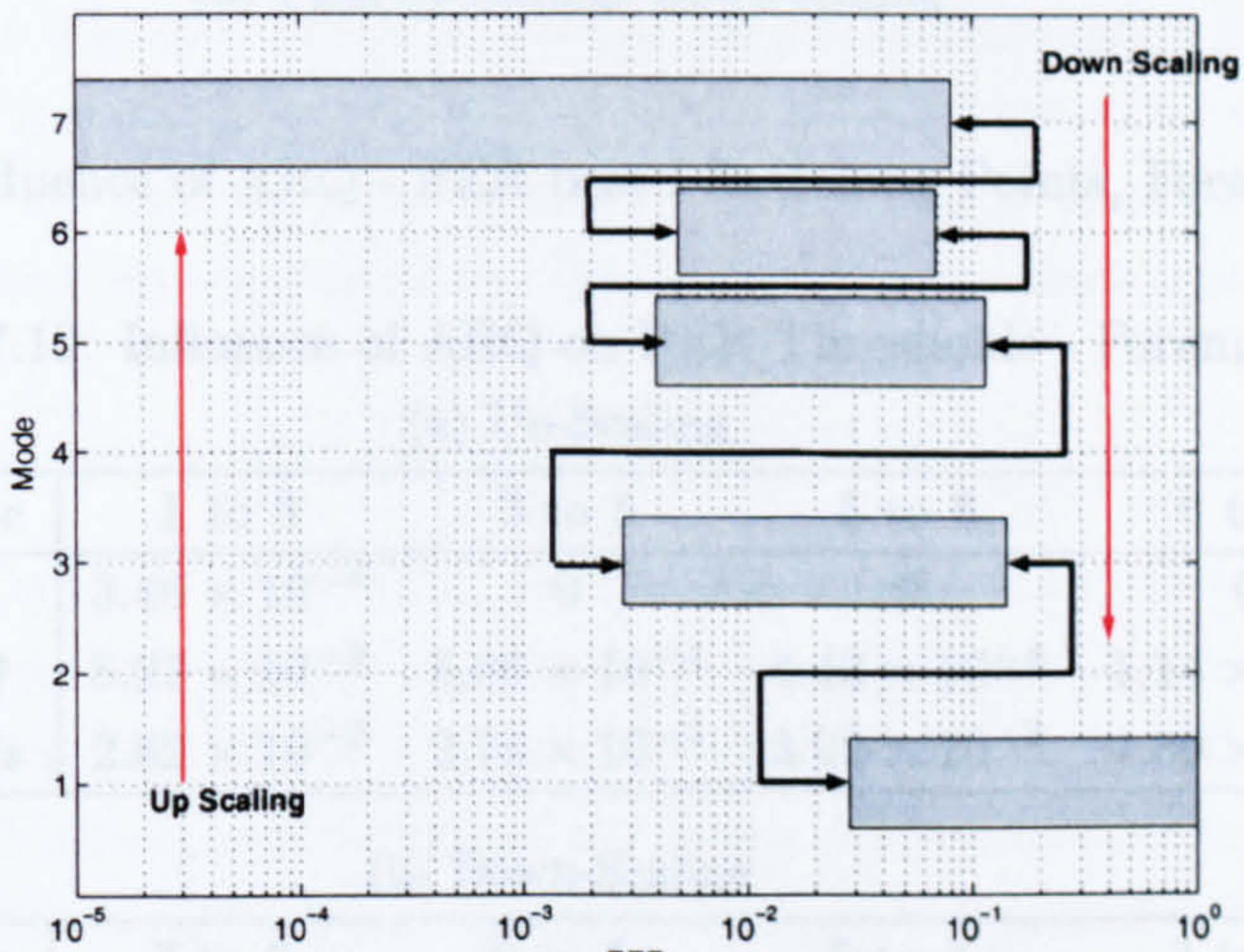
Figure 7.16: *Foreman* Sequence, Influence of ARQ on PSNR



(a) No ARQ

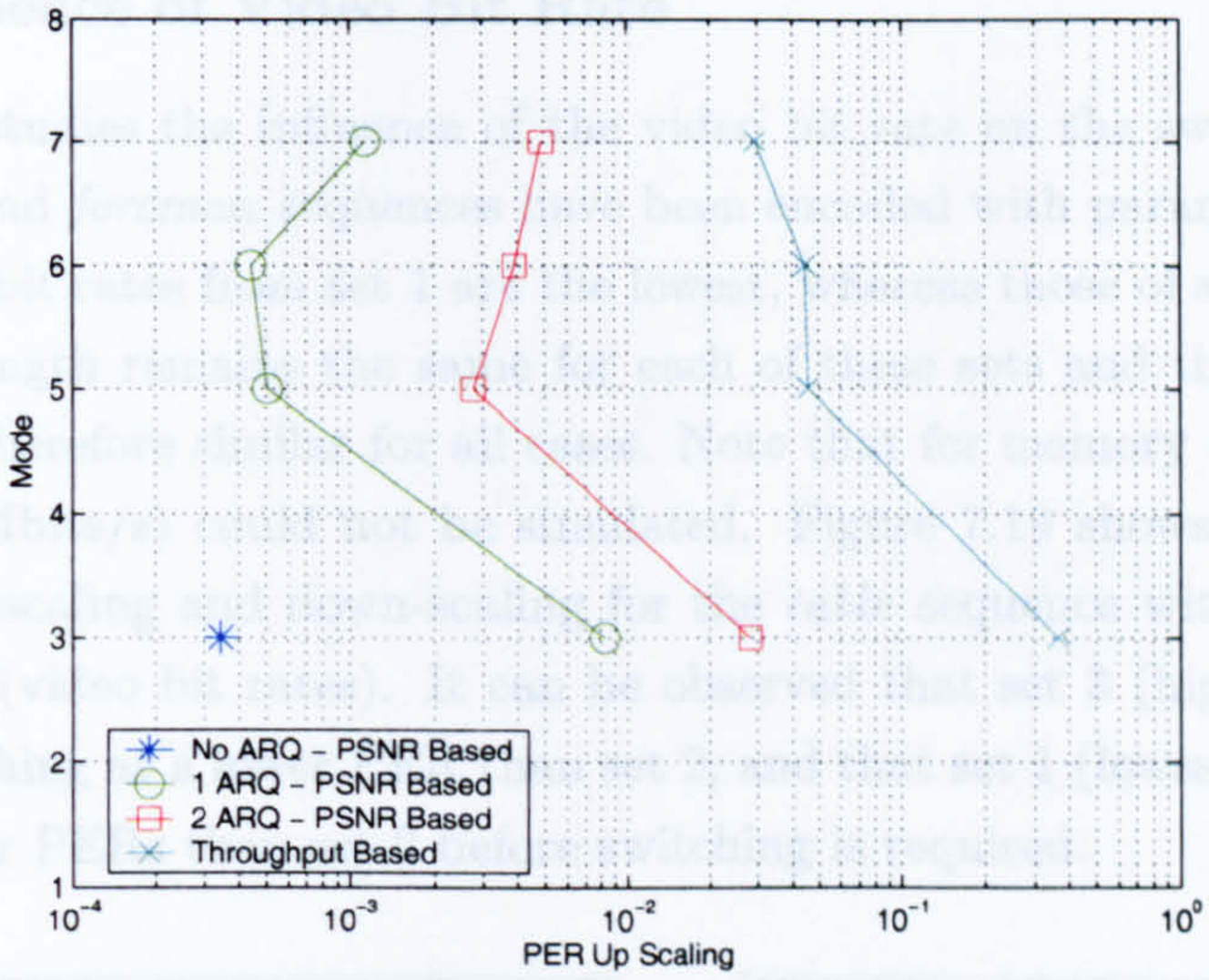


(b) One ARQ

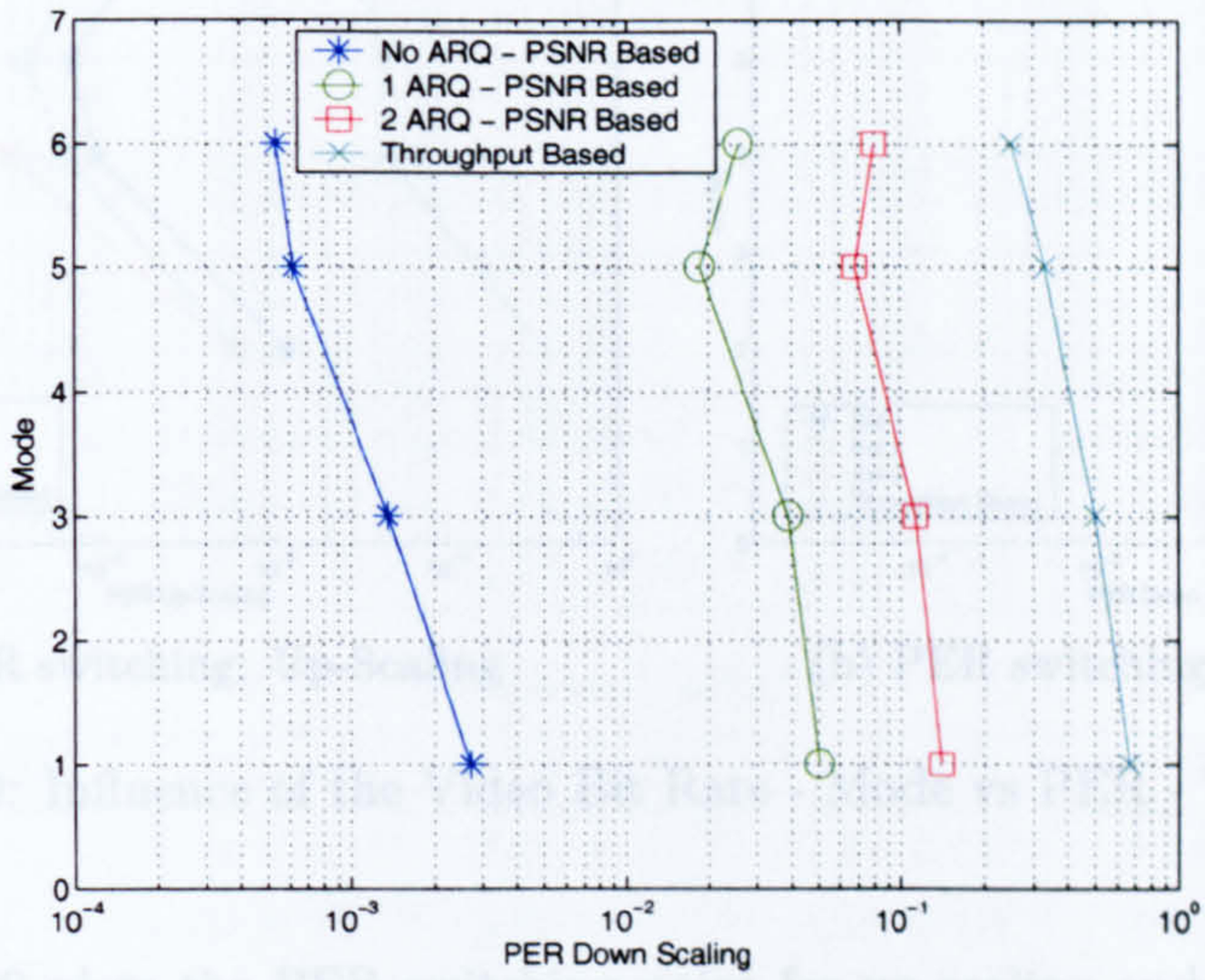


(c) Two ARQs

Figure 7.17: *Foreman* Sequence, Influence of ARQ on PER switching



(a) PER Switching: Up-scaling



(b) PER Switching: Down-scaling

Figure 7.18: Influence of ARQ - PER based Switching Points, Foreman, Set 2

Table 7.12: Influence of ARQ on PER Thresholds - Foreman

(a) Up-Scaling

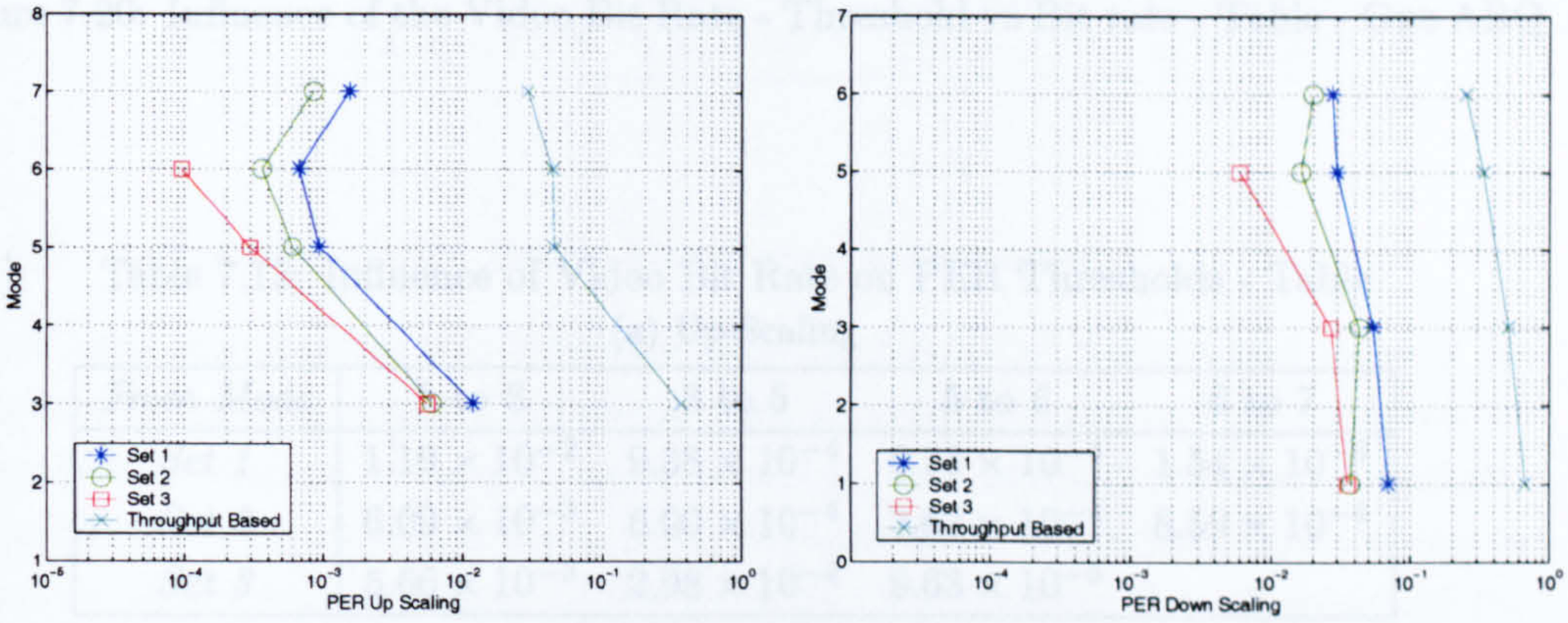
From Mode	1 to 3	3 to 5	5 to 6	6 to 7
No ARQ	3.46×10^{-4}	0	0	0
One ARQ	8.27×10^{-3}	5.06×10^{-4}	4.43×10^{-4}	1.14×10^{-3}
Two ARQs	2.82×10^{-2}	2.78×10^{-3}	3.92×10^{-3}	4.89×10^{-3}

(b) Down-Scaling

From Mode	7 to 6	6 to 5	5 to 3	3 to 1
No ARQ	2.72×10^{-3}	6.22×10^{-4}	1.36×10^{-3}	5.36×10^{-4}
One ARQ	2.55×10^{-2}	1.85×10^{-2}	3.88×10^{-2}	5.04×10^{-2}
Two ARQs	7.97×10^{-2}	6.82×10^{-2}	1.13×10^{-1}	1.40×10^{-1}

7.4.4 Influence of Video Bit Rate

This section studies the influence of the video bit rate on the switching points. The *Akiyo*, *table* and *foreman* sequences have been encoded with parameters from sets 1, 2 and 3. Video bit rates from set 1 are the lowest, whereas those of set 3 are the highest. The packet length remains the same for each of these sets and the throughput based algorithm is therefore similar for all cases. Note that for memory usage reasons, mode 7 of set 3 (9Mbits/s) could not be simulated. Figure 7.19 shows the PER switching points for up-scaling and down-scaling for the *table* sequence with one ARQ and for all three sets (video bit rates). It can be observed that set 3 (highest video bit rate) requires switching at a lower PER than set 2, and that set 1 (lowest video bit rate) can tolerate higher PERs than set 2 before switching is required.



(a) PER switching: Up-Scaling (b) PER switching: Down-scaling

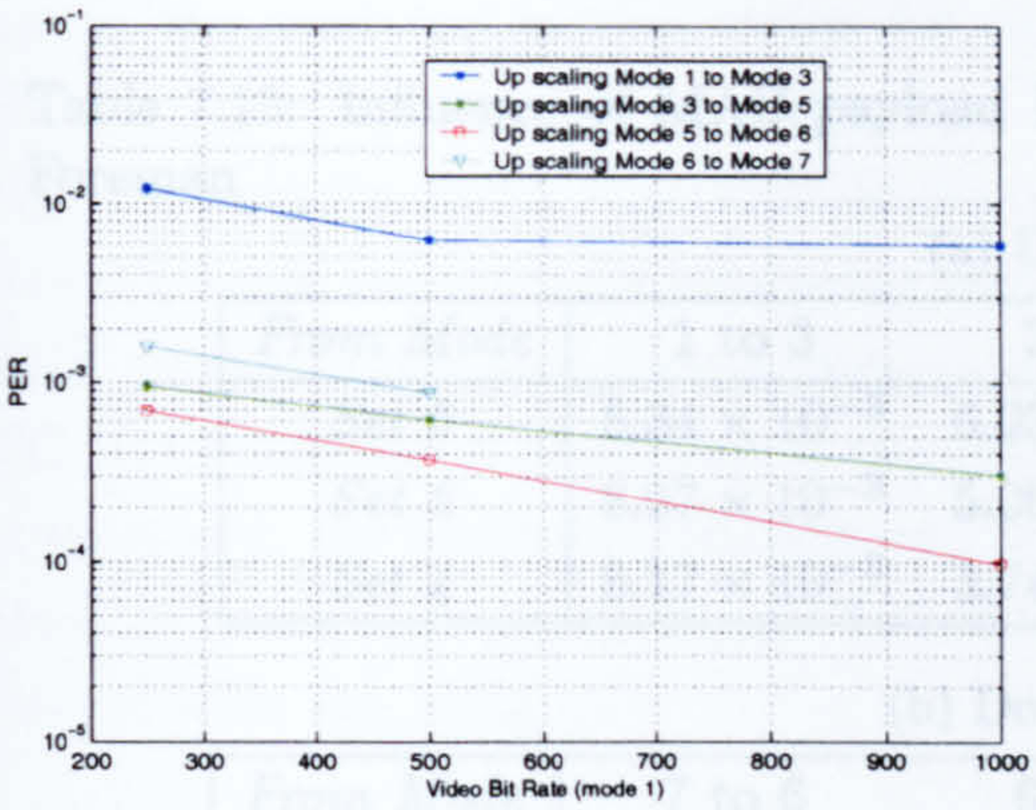
Figure 7.19: Influence of the Video Bit Rate - Mode vs PER - Table - One ARQ

Figure 7.20 plots the PER switching point for up-scaling and down-scaling versus the video bit rate (the first video bit rate of the set corresponding to mode 1). As the video bit rate increases, i.e. as the quality of the transmitted video increases, the PER switching points are seen to occur at lower values. This is summarised in table 7.13. Similar results were obtained with the *foreman* and *akiyo* sequences.

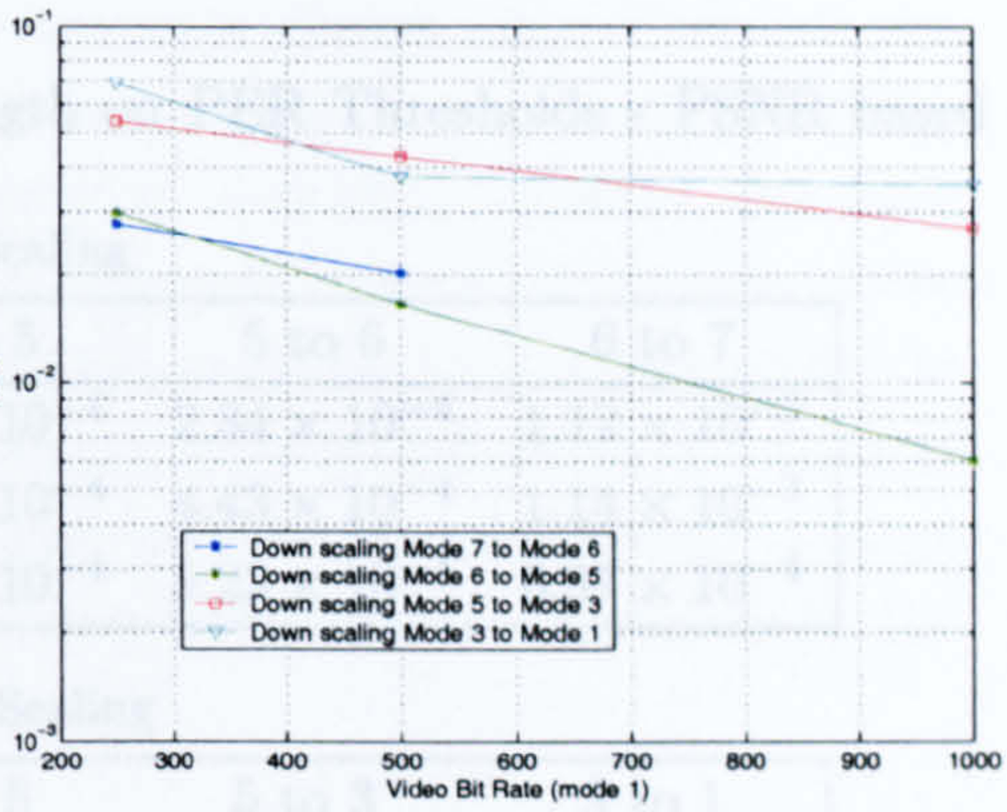
7.4.5 Influence of MAC payload length

In this section, the effect of the MAC payload length on the switching points is studied. *Akiyo*, *table* and *foreman* sequences have been encoded with parameters from sets 5, 2 and 4. Packet lengths from set 5 are the smallest, whereas packet lengths from set 4 are the largest. Video bit rates remain the same. As the packet lengths are different, the PERs for a given C/N differ. However changes in thresholds for the throughput based algorithm are relatively insignificant, especially at such high PER as shown in table 7.14.

Table 7.15 shows the PER switching points of the proposed scheme for up and down-scaling for the foreman sequence with one ARQ and for the different sets (packet



(a) PER switching: Up-scaling



(b) PER switching: Down-scaling

Figure 7.20: Influence of the Video Bit Rate - Threshold vs Bit rate - Table - One ARQ

Table 7.13: Influence of Video Bit Rate on PER Thresholds - Table

(a) Up-Scaling

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
Set 1	1.19×10^{-2}	9.38×10^{-4}	6.78×10^{-4}	1.54×10^{-3}
Set 2	6.09×10^{-3}	6.00×10^{-4}	3.63×10^{-4}	8.59×10^{-4}
Set 3	5.66×10^{-3}	2.98×10^{-4}	9.63×10^{-5}	-

(b) Down-Scaling

From Mode	7 to 6	6 to 5	5 to 3	3 to 1
Set 1	2.74×10^{-2}	2.96×10^{-2}	5.35×10^{-2}	6.84×10^{-2}
Set 2	1.99×10^{-2}	1.63×10^{-2}	4.26×10^{-2}	3.72×10^{-2}
Set 3	-	5.98×10^{-3}	2.67×10^{-2}	3.54×10^{-2}

Table 7.14: Influence of MAC payload Length on PER Thresholds - Throughput based

(a) Up-Scaling

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
Set 5	3.35×10^{-1}	4.50×10^{-2}	4.48×10^{-2}	3.22×10^{-2}
Set 2	3.67×10^{-1}	4.56×10^{-2}	4.44×10^{-2}	2.94×10^{-2}
Set 4	3.86×10^{-1}	4.47×10^{-2}	5.57×10^{-2}	2.51×10^{-2}

(b) Down-Scaling

From Mode	7 to 6	6 to 5	5 to 3	3 to 1
Set 5	2.63×10^{-1}	3.29×10^{-1}	4.95×10^{-1}	6.42×10^{-1}
Set 2	2.50×10^{-1}	3.31×10^{-1}	5.01×10^{-1}	6.64×10^{-1}
Set 4	2.39×10^{-1}	3.46×10^{-1}	5.14×10^{-1}	6.85×10^{-1}

lengths). This shows that the packet length does not influence considerably the thresholds. Similar results have been obtained with the *table* and *akiyo* sequences.

Table 7.15: Influence of MAC payload Length on PER Thresholds - PSNR based - Foreman

(a) Up-Scaling

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
Set 5	8.34×10^{-3}	6.03×10^{-4}	2.34×10^{-4}	1.12×10^{-3}
Set 2	8.27×10^{-3}	5.06×10^{-4}	4.43×10^{-4}	1.14×10^{-3}
Set 4	8.17×10^{-3}	3.74×10^{-4}	5.23×10^{-4}	6.37×10^{-4}

(b) Down-Scaling

From Mode	7 to 6	6 to 5	5 to 3	3 to 1
Set 5	2.29×10^{-2}	1.58×10^{-2}	3.77×10^{-2}	5.11×10^{-2}
Set 2	2.55×10^{-2}	1.85×10^{-2}	3.88×10^{-2}	5.04×10^{-2}
Set 4	2.15×10^{-2}	2.81×10^{-2}	3.92×10^{-2}	4.73×10^{-2}

7.4.6 Influence of the Error Resilience techniques

Error resilience features will determine how the quality of the received video is influenced by channel errors. Techniques such as pre-processing error concealment as described in chapter 4 (layered/scalable coding with unequal protection, multiple-description coding, insertion Intra MBs, Forward Error Correction, and the use of Flexible Macroblock Ordering (FMO) in H.264) will improve the robustness of the video. At the decoder, post-processing concealment techniques [11, 102] will try to recover and minimise the visual artifacts of damaged and/or missing MBs as detailed in chapter 4.

In this section, the effects of two post-processing error concealment algorithms are compared. In the current scenario, one MAC packet contains one video slice (plus the header). Any packet in error is re-transmitted at the MAC layer (with the ARQ mechanism) until the retry-limit is reach. If after the number of ARQs allowed, the packet has not been received correctly, the packet is dropped and the video decoder assumes that the slice has been lost. The decoder must therefore process the missing data to reduce visual artifacts.

The basic and simplest error concealment algorithm replaces the missing MB by the corresponding MB in the previous frame for INTER concealment and weight-averages of the neighboring MBs for INTRA concealment [91]. This is the *Previous Frame Copy* (PFC) algorithm. The second concealment algorithm is the *Advanced Error Concealment* (AEC) algorithm and is defined in [133] and [16]. It has been adopted as the reference framework on which further concealment enhancement in the H.264 JVT standard would be based. Further details on error resilience and concealment techniques can be found in section 4.4.4.

Parameters of set 2 have been used to encode *akiyo*, *foreman*, *table* and *stefan*.

Table 7.16 shows the switching points for up-scaling modes for the four sequences and compares the two algorithms when one ARQ is used. As *akiyo* contains very little motion, the predicted motion vector for compensation is always zero. Therefore the AEC and the PFC algorithms present the same characteristics. Switching points for *akiyo* are then the same, regardless of the concealment algorithm. However, the three other sequences feature more significant motion and the AEC algorithm provides better PSNR results. Mode switching occurs at higher PER with the AEC compared to the PFC. The concealment algorithm at the video decoder therefore influences the choice of the threshold: the better the concealment, the higher the PER the sequence can support before switching is required. Figure 7.21 shows the switching points comparison for the *foreman* with one ARQ.

Table 7.16: Influence of Error Concealment on PER Thresholds Up-Scaling- One ARQ - Set 2

(a) Akiyo

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
PFC	8.16×10^{-3}	5.29×10^{-4}	4.88×10^{-4}	1.48×10^{-3}
AEC	8.16×10^{-3}	5.29×10^{-4}	4.88×10^{-4}	1.48×10^{-3}

(b) Foreman

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
PFC	6.47×10^{-3}	2.46×10^{-4}	3.41×10^{-4}	7.53×10^{-4}
AEC	8.27×10^{-3}	5.09×10^{-4}	4.43×10^{-4}	1.14×10^{-3}

(c) Table

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
PFC	5.51×10^{-3}	3.81×10^{-4}	2.11×10^{-4}	6.22×10^{-4}
AEC	6.09×10^{-3}	6.00×10^{-4}	3.63×10^{-4}	8.59×10^{-4}

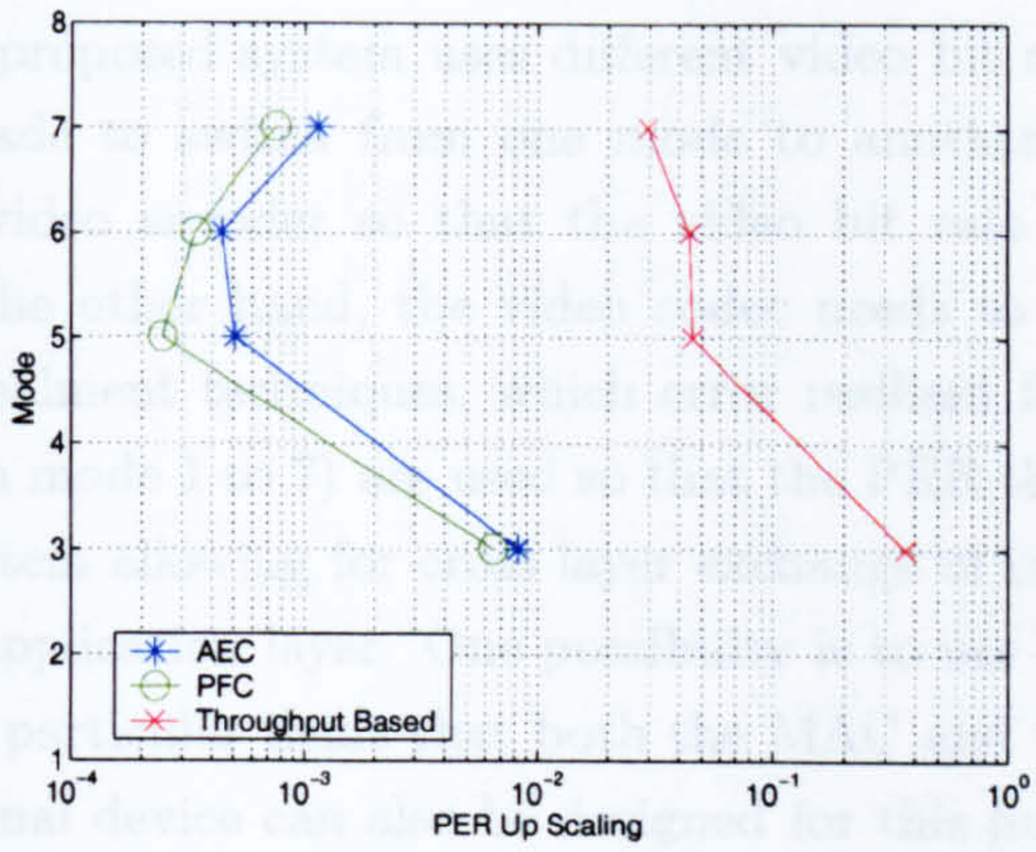
(d) Stefan

From Mode	1 to 3	3 to 5	5 to 6	6 to 7
PFC	7.68×10^{-3}	4.65×10^{-4}	4.00×10^{-4}	1.00×10^{-3}
AEC	1.04×10^{-2}	6.40×10^{-4}	5.98×10^{-4}	1.33×10^{-3}

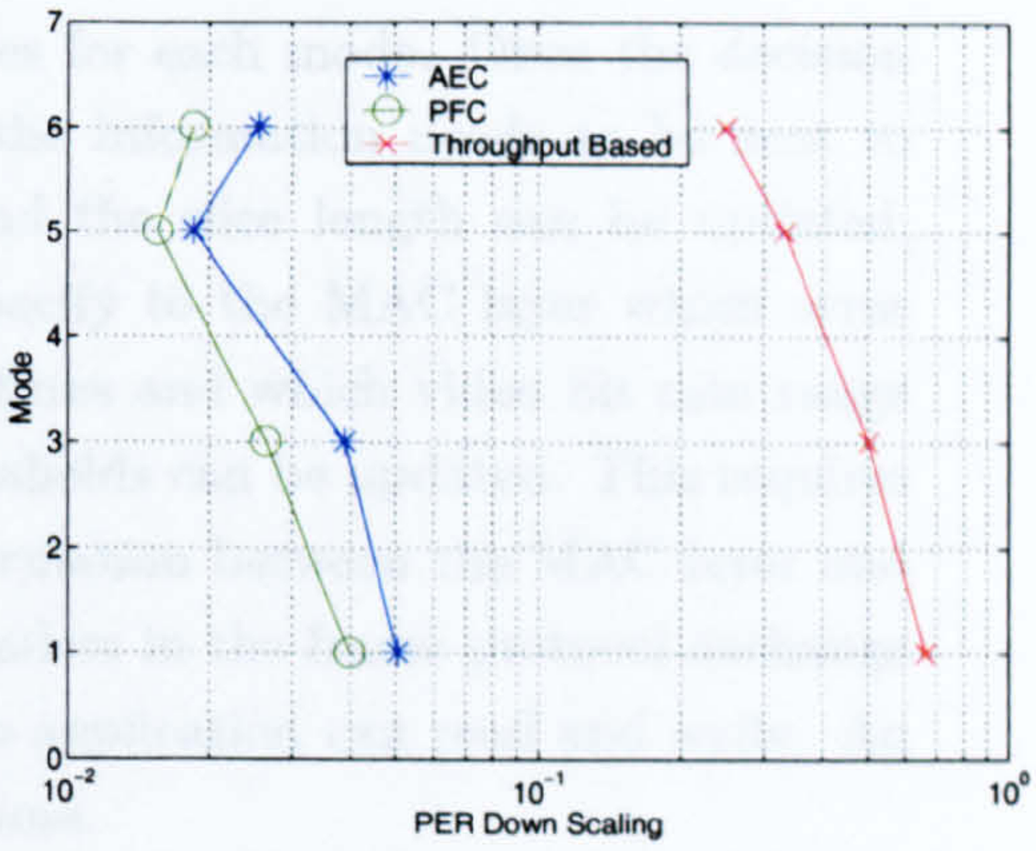
Strong concealment algorithms would be then very desirable when using off the shelf WLAN cards and link adaptation algorithms. Without, such concealment, the standard link adaptation will result in considerable video distortion unless the link adaptation algorithm is adjusted as mentioned earlier.

7.4.7 Implementation Issues

In this section, practical issues and recommendations for the implementation of the proposed algorithm are presented. The proposed approach uses PHY PER thresholds that have been shown to be much lower than traditional mechanisms. Results were shown in an empirical manner, stressing the need for an appropriate algorithm designed



(a) PER switching: Up-scaling



(b) PER switching: Down-scaling

Figure 7.21: Influence of the Concealment technique - Foreman - One ARQ

to optimise the received video quality rather than the throughput. The derivation of rigorous PER thresholds with an optimising algorithm is therefore not in the scope of this thesis and is left for future investigations.

7.4.7.1 Latency Requirements

In the case of a low latency system, per-packet adaptation is possible. Feedback information is available for each packet [48]. Per-packet adaptation is however not realistic nor necessary for video transmission. As the proposed scheme is based on the PER, the estimation of PER is critical. It is either estimated at the transmitter or alternatively at the receiver with the information being sent back to the transmitter. One solution is the use of C/N based look-up tables. The receiver and the transmitter have knowledge of the Received Signal Strength Indication (RSSI) and the PER can be computed from this value [181]. The main draw-back of this method is the reliability of the C/N value and the variation of PER with channel type [26]. In this case, a hysteresis (or guard zone) is required around the PER thresholds so that the system does not oscillate between two adjacent modes. To prevent oscillation, the system might have to stay in the lower mode for a certain minimum time duration. Another solution is to estimate the PER through a decision window. For example, acknowledged packets at the transmitter (or corrupted packets at the receiver) are counted during this decision window and the PER is then computed. The size of the decision window will depend on how fast the radio channel is changing. For a slow changing radio channel where users are not moving, fast adaptation is not required. In this case, adapting the operating link mode at a rate of around once every few seconds is reasonable. It is not needed to update the transmission rate more often under steady channel conditions. This will allow the decision window to be large compared to packet duration and hence the gathered packet loss statistics can be accurate.

7.4.7.2 Design

The proposed system uses different video bit rates for each mode. Once the decision is made to switch from one mode to another, the information needs to be sent to the video encoder so that the video bit rate and the slice length can be updated. On the other hand, the video codec needs to specify to the MAC layer which error concealment techniques, which error resilient features and which video bit rate range (from mode 1 to 7) are used so that the PER thresholds can be updated. This requires a system allowing for cross layer exchange of information between the MAC layer and the application layer. One possibility is to use headers in the frame protocol exchange with particular fields that both the MAC and the application can read and write. An external device can also be designed for this purpose.

In the case of a slow changing environment, another possible design for the studied algorithm is to use PER threshold tables.The implementation would progress as follows; assuming the MAC knows which threshold table to use (video bit rate, initial packet length, number of ARQs, concealment technique):

1. Over a decision window of 5-10 seconds, at the transmitter, count the number transmitted packets N_t and the number of ACKs received N_{ack}.

2. Compute the PER over this decision window:

$$PER = \frac{(N_t - N_{ack})}{N_t} \tag{7.4}$$

3. Compare the PER with $PER_u(m)$ and $PER_d(m)$, the up and down-scaling thresholds respectively, where m represents the current mode:

   - If ($PER \leq PER_u(m)$ and $i \neq 7$), up-scale to mode ***m+1***
   - If ($PER_d(m) \leq PER$ and $i \neq 1$), down-scale to mode ***i-1***
   - Stay in mode m otherwise.

A pragmatic solution would be for the video codec to supply the appropriate switching thresholds to the MAC layer. If this is not possible, then default values will be used.

7.5 Conclusion

In this chapter, a link adaptation strategy designed for H.264 video transmission over the IEEE 802.11 MAC and IEEE 802.11a PHY layer has been investigated and characterised. Emphasis has been given to robust transmission to several handheld devices with no (or limited) retransmissions. Previous algorithms focused on maximising the error free data throughput. The study was motivated by the fact that these algorithms do not take into account the nature of the content and the low delay requirements for time-bounded applications. Since each transmission mode allows different nominal bit rates with different reliability, the investigated algorithm transmits the same video sequence at different video bit rates on each mode in a scalable manner. Video and

MAC layers are required to exchange information so that each layer can up-date their respective parameters. This approach acknowledges that error resilient video codecs are capable of tolerating errors, up to a certain level. Moreover, it optimises the overall received video quality rather than the throughput. Since ideal adaptation based on received PSNR is not feasible, a more practical approach was proposed based on PER thresholds. The scheme is designed for low latency video transmission without strong reliance on ARQ and switches at far lower PERs compared to the classic throughput based algorithm. Empirical results show that, practically, throughput based and traditional PER based algorithms would switch down for PHY error rates greater than or equal to 0.1, whereas the proposed algorithm down-scales the mode for a PER of around 5×10^{-2}. Similarly, the throughput based algorithms would switch up for a PER of around 5×10^{-2}, whereas the presented scheme up-scales the operating mode for a PER of around 5×10^{-4}. Results shows that, since they use inappropriate thresholds, traditional link adaptation mechanisms deliver poor video quality when applied to a time-bounded video transmission requiring no (or limited) retransmissions. A gain of 7 dB in PSNR was observed with the investigated algorithm, compared to the traditional throughput-based algorithm. The choice of up and down-scaling thresholds is therefore very critical for reliable low latency.

Different parameters that influence the switching thresholds were studied. A good concealment algorithm allowed switching at higher PERs, and with strong concealment the traditional data throughput thresholds were seen to become usable. Similarly, as the number of layer 2 ARQs increases, switching points were seen to move towards higher PERs and approached the throughput-based case.

Chapter 8

Measurements and Performance Analysis

This chapter investigates and studies video transmission over practical IEEE 802.11a/g wireless LANs. Section 8.1 introduces the chapter and describes the platform used in order to collect data. Processing of the measurement data is conducted in section 8.2. Loss patterns extracted from measurements are used to simulate video transmissions. This section includes a comparison of a UDP and a TCP link transmission over IEEE 802.11g, a study of the effect of the packet length. The two studies developed in chapter 6 on packetisation strategies and in chapter 7 on Link Adaptation are also detailed. Finally section 8.3 concludes this chapter.

8.1 Introduction

Wireless LANs are now widely available in the market place. At the time of the writing, IEEE 802.11a at 5GHz is not available in Europe. However, IEEE 802.11b at 2.4GHz is widely available and deployed in public and private buildings, in schools, universities and institutions. Recently, IEEE 802.11g at 2.4GHz has been launched, enhancing the PHY bit rate relative to the 802.11b. If the technology follows the specifications of IEEE 801.11 and IEEE 802.11g [3, 39], there are still a number of points that are left to manufacturers, where implementations are proprietary choices and solutions, and depending on the application, these can be very critical. The following items, that are not specified by standards, should be highlighted when designing a real-time video transmission system over video WLAN.

- The key feature of IEEE 802.11g is its backward compatibility with 802.11b. This means that the 802.11b MAC parameters have to be applied, resulting in lower throughput efficiency and larger delays (see chapter 3). IEEE 802.11g can operate in two mode. In b-backward compatible mode, performances are reduced. IEEE 802.11a performance levels can be obtained only when no 802.11b compatibility is applied.

- The choice of TCP/UDP determines the type of applications that the system

can support: time bounded and delay sensitive (for UDP) and time and delay independent (for TCP).

- The maximum number of retries at the MAC layer determines the inherent delay of the system (see chapter 3).

- Encapsulation: the way video packets are transmitted to the underlying networks determines the throughput performance of the system (see chapter 6).

- Link Adaptation: the way the system switches from one mode to another one depends on the application, its properties and its requirements. It determines the overall performance of the system (see chapter 7).

In order to develop an understanding of the relevant influence of certain key parameters, real data measurements have been conducted between two laptops, either static or mobile. The IEEE 802.11a and IEEE 802.11g/b performance was assessed with real cards via a client-server software tool developed by ProVision Communications Limited, Bristol, UK, in the context of the FP6 WCAM European Project. The software allows to log the PER and delays taken at the top of the TCP or UDP layer and the RSSI and the link speed taken at the PHY layer. The software provides therefore a powerfull tool for a cross-layer analysis of the transmission for both UDP and TCP transmissions. A full description of the software is given in the appendix F. Logged data is used in order to generate loss patterns that can be used to simulate off-line video transmission over wireless LANs.

8.1.1 Platform Description

In this section, the platform used for the measurements is described. The platform consists of a client/server software pair running on two WindowXP-based laptops, where two PCMCIA Cardbus IEEE 802.11b/g or IEEE 802.11a devices are mounted as shown in figure 8.1 [182].

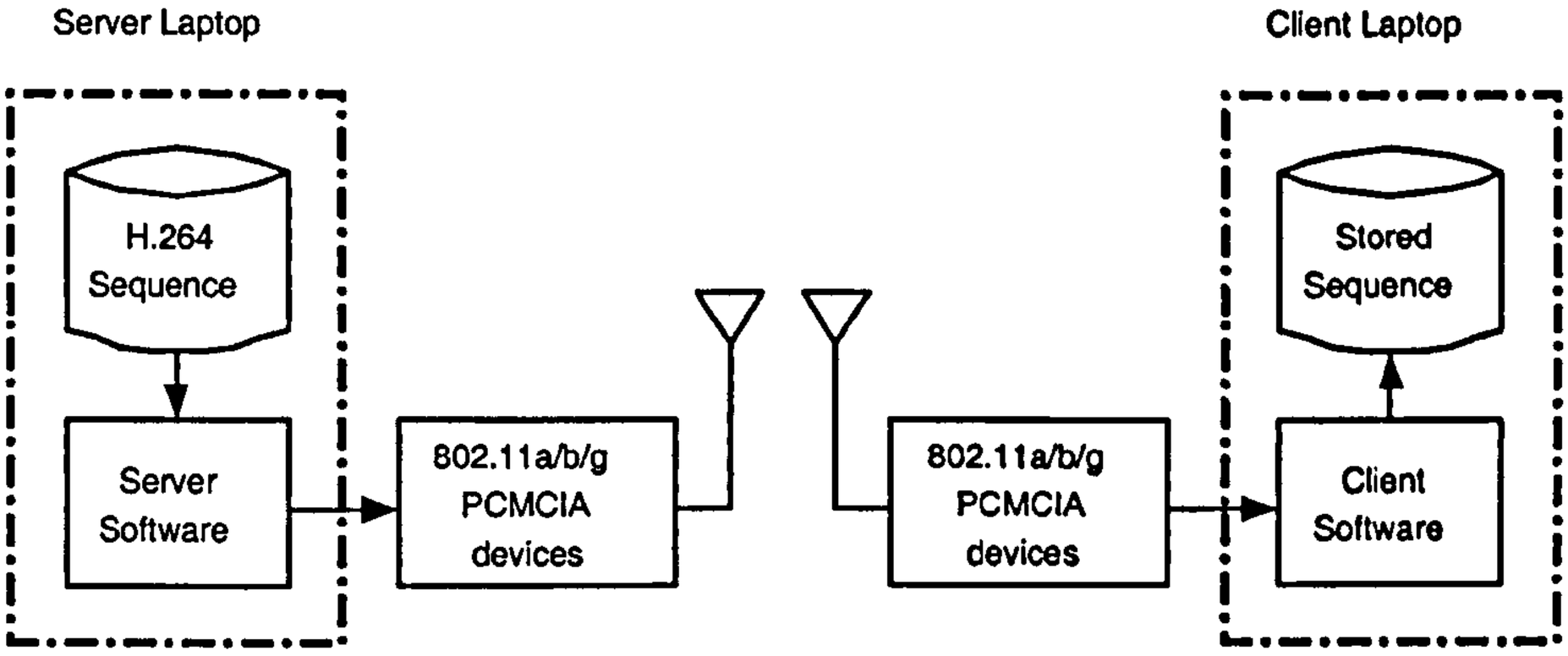


Figure 8.1: Platform Set up

8.1.2 Hardware

The PCMCIA Cardbus devices used were:

- F5D 7010 Belkin 54G IEEE 802.11b/g: IEEE 802.11g chipsets back compatible with 802.11b manufactured by $Belkin^{TM}$ as shown in figure 8.2 [182].

- WLI - CB - G54A Buffalo 802.11b/g: IEEE 802.11g chipset backward compatible with 802.11b manufactured by $Buffalo^{TM}$ as shown in figure 8.2 [182].

- AIR-CB21 CISCO: IEEE 802.11a chipsets manufactured by $CISCO^{TM}$

The platform comprises two laptops. Hence, no collisions occur with other stations. The only possible cause for retransmissions is due to channel errors. The two cards are connected in the form of an ad-hoc network.

Figure 8.2: IEEE 802.11g configuration at 2.4GHz

8.2 Measurements

This section details measurements and H.264 video transmission simulations implemented using error patterns generated from the practical logged cross-layer data measurements (see appendix F). The video codec uses the Advanced Error Concealment (AEC) algorithm presented in chapter 4. Note that, because of the lack flexibility of the available cards, the maximum number of retransmission is not manually tuneable and is fixed to 32 by the manufacturers.

8.2.1 TCP/UDP transmission comparisons

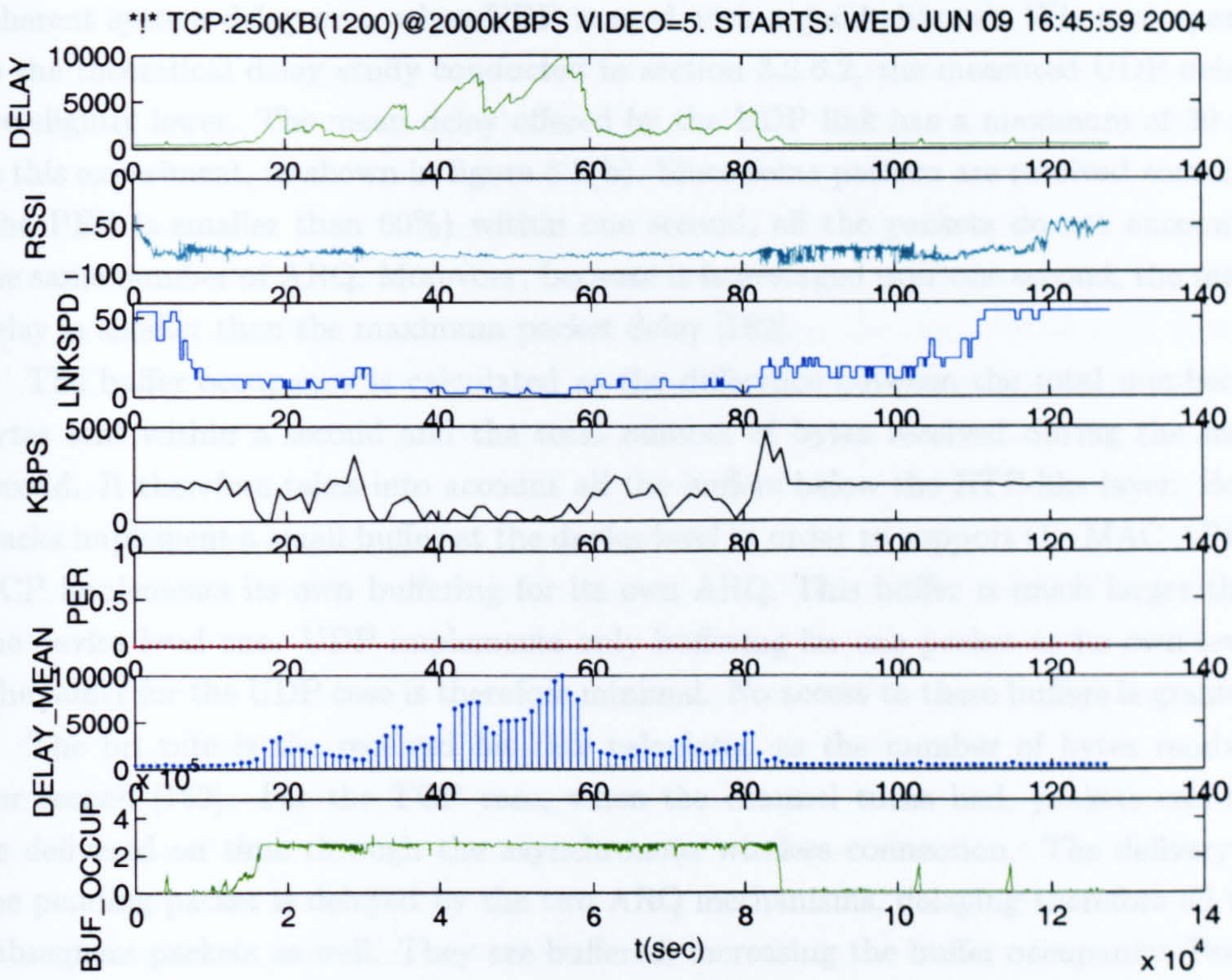
In this section, transmissions over TCP and UDP links are studied. For these measurements, the IEEE 802.11g $Buffalo^{TM}$ card was used on the server. The client is

mounted with the IEEE 802.11g $Belkin^{TM}$ card. Note that these two cards are backward compatible with IEEE 802.11b. The mobile client moves away from and back to the static server at a walking pace in a Line-of-Sight (LOS) outdoor environment over a period of 140 seconds. During this experiment, cards were pointing away from each other. The mode selection is left up to the cards internal algorithm, which is unfortunately unknown. The maximum number of retries at the MAC is set by the card and is fixed to 32. The target bit rate is set to 2000 kbits/s. TCP is implemented with a buffer of 125ms (250 kbits). The packet size is 1200 bytes. Figure 8.3 compares the performance of transmission with both UDP and TCP links [182].

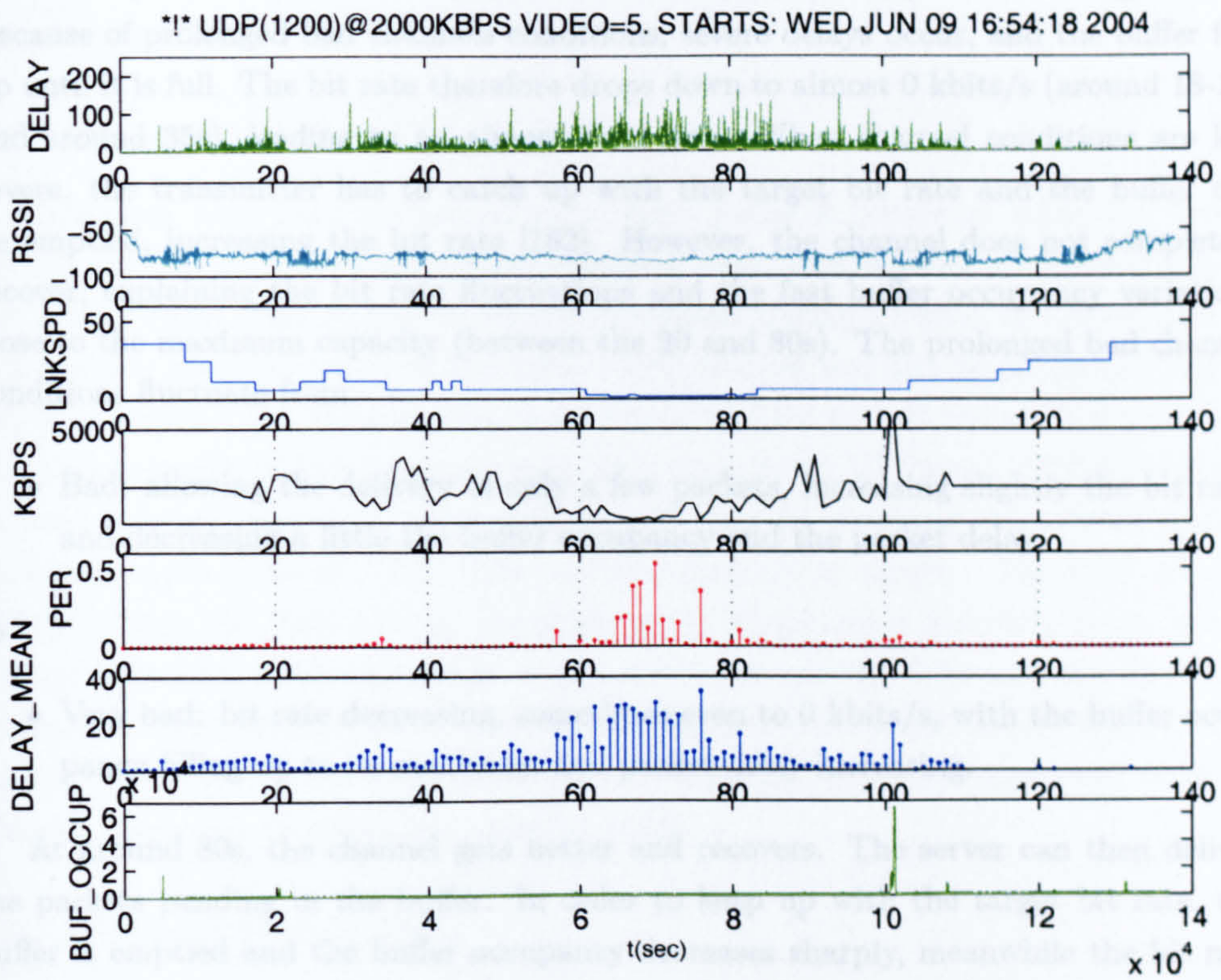
For both cases, as the mobile terminal moves away from the server, the RSSI decreases from -50 dBm to -90 dBm. The link speed varies, showing the link adaptation algorithm developed by the manufacturers. The link speed reduces as the mobile moves away, and then increases on returning to the server. This IEEE 802.11g card is backward compatible with IEEE 802.11b and can not be locked to 802.11g only. IEEE 802.11b has 4 operating modes (1, 2, 5.5 and 11 Mbits/s), to be added to the 8 already existing in IEEE 802.11g. However, the server card does not use the 6, 9 and 12 Mbits/s modes of the IEEE 802.11g card, but directly switches down to the four IEEE 802.11b modes. The number of link speeds available is nine and the speeds are: 1, 2, 5.5, 11, 18, 24, 36, 48 and 54Mbits/s [182]. Note that due initialisation problem, the calculated UDP link starts with mode 6 at 36Mbits/s and never moves to the highest rate (54 Mbits/s).

The *PER* at the application layer is calculated by checking the received packet counter at the RTP-like layer. If after the maximum number of retries at the MAC, a packet is still not acknowledged, the MAC drops it. If UDP is used, no retransmission is attempted and the packet is then missing at the application layer. If TCP is used, the TCP retransmission mechanism takes over and retries to retransmit the packet until the packet is received correctly. It therefore re-enters the retransmission process at the MAC. TCP offers error-free transmission, whereas UDP presents lost packets when the client moves far from the server as shown in figure 8.3. It can be seen that the link is almost dead when the UDP PER reaches almost 60%. Note: with the use of UDP, the PER after the MAC is the same as the PER at the application layer.

Delay represents the packet delay plotted against time, whereas *Delay_Mean* represents the average of the packet delay within one second. Both are expressed in ms. Although TCP allows error-free transmission, it does however introduce large packet delays (up to 10000 ms (10s) in this experiment), as shown in figure 8.3(a). The mean delay can be very high. These considerable delays are due to the TCP ARQ mechanism which accumulates the MAC ARQ delays at each TCP retransmission until the packet is received correctly. On the other hand, in this experiment, UDP presents only a maximum of packet delay of 250 ms, as shown in figure 8.3(b), which is entirely due to the MAC ARQ, occurring when the channel is bad. Note that before each session, the client sends messaging information using TCP packets in order to trigger clocks. TCP ARQ is used until this information is received correctly, and hence this packet may be



(a) TCP



(b) UDP

Figure 8.3: UDP/TCP Route Comparison, IEEE 802.11g, backward compatible with 802.11b

delayed. The client and server clock are therefore not synchronised. This explains the inherent system delay even when UDP is used with a good channel. When compared to the theoretical delay study conducted in section 3.2.6.2, the measured UDP delays are slightly lower. The mean delay offered by the UDP link has a maximum of 30 ms in this experiment, as shown in figure 8.3(b). Since some packets are received correctly (the PER is smaller than 60%) within one second, all the packets do not encounter the same number of ARQ. Morevoer, because it is averaged over one second, the mean delay is smaller than the maximum packet delay [182].

The buffer occupancy is calculated as the difference between the total number of bytes sent within a second and the total number of bytes received during the same second. It therefore takes into account all the buffers below the RTP-like layer. Both stacks implement a small buffer at the device level in order to support the MAC ARQs. TCP implements its own buffering for its own ARQ. This buffer is much larger than the device level one. UDP implements only buffering for one packet at its own level. The buffer for the UDP case is therefore minimal. No access to these buffers is granted.

The bit rate is the received bit rate calculated as the number of bytes received per second [182]. For the TCP case, when the channel turns bad, packets can not be delivered on time through the asynchronous wireless connection. The delivery of the pending packet is delayed by the two ARQ mechanisms, delaying therefore all the subsequent packets as well. They are buffered, increasing the buffer occupancy. Fewer packets are received within one second, the packet delay builds up and the bit rate decreases (between the 15th and the 18th seconds of the experiment of figure 8.3). Because of prolonged bad channels conditions, severe delays occur, and the buffer fills up until it is full. The bit rate therefore drops down to almost 0 kbits/s (around 18-19s and around 35s), leading to an almost dead link. When channel conditions are less severe, the transmitter has to catch up with the target bit rate and the buffer can be emptied, increasing the bit rate [182]. However, the channel does not completely recover, explaining the bit rate fluctuations and the fast buffer occupancy variations close to the maximum capacity (between the 20 and 80s). The prolonged bad channel conditions fluctuate from:

- Bad: allowing the delivery of only a few packets, increasing slightly the bit rate, and decreasing a little the buffer occupancy and the packet delay,

to

- Very bad: bit rate decreasing, sometimes even to 0 kbits/s, with the buffer occupancy filling up to its maximum and packet delay increasing.

At around 80s, the channel gets better and recovers. The server can then deliver the packets pending in the buffer. In order to keep up with the target bit rate, the buffer is emptied and the buffer occupancy decreases sharply, meanwhile the bit rate strongly increases until steady state good channel conditions are reached, allowing packet delivery without excessive delay and at the target rate.

In the UDP case of figure 8.3, the target bit rate is kept until the channel conditions degrade. When the channel gets bad, since there is no ARQ and no buffering at the UDP level, packets are dropped because either the device level buffer is full or they have not been received correctly after the MAC ARQs. The buffering is however minimal and the induced delays are small. A peak in the buffer occupancy occurs when the channel conditions turn from bad to not so bad [182]. Under good conditions, packets are transmitted without, or with very few, retransmissions. Under bad conditions, packets are not transmitted because the ARQ mechanism can not ensure error-free delivery. The bit rate therefore decreases. However, under intermediate conditions, packets are transmitted with many retries and with a 'larger' delay, explaining the packet delays even at a relatively low PER. Because the buffer is minimal, emptying is almost instantaneous, creating peaks in the bit rate.

With the TCP case of figure 8.3, due to the excessive delays, the jitter buffer at the receiver should be large enough to smoothen out excessive delay jitter [182]. For low latency applications, such as real-time video transmission, TCP is not appropriate and UDP transfer is faster than ARQ-based TCP transmission [137].

When UDP is used, the application layer can not control and ensure that packets will not be dropped at the UDP level because of UDP buffer overflow due to bad channel conditions. This is because UDP does not have enough flexibility and it is not possible for the application to probe the UDP buffer to check whether it is full or not. This is a weakness of UDP. One solution is to allow low level MAC access in order to check whether the buffer at the MAC is capable of accepting another packets or not before sending the next UDP packet.

8.2.2 Video Simulation Conditions

The simulations have been performed using error patterns generated using the two IEEE 802.11a $CISCO^{TM}$ cards. Unless specified, both client and server are static in a Non Line-of-Sight (NLOS) indoor office environment during a period of 120 seconds. The automatic mode selection of the card is switched off and the 16 QAM 1/2 rate (mode 5) is chosen. The *foreman* video sequence is encoded with H.264 at 1000 kbits/s with a packet length of 150 bytes and with an I frame every 90 frames. PSNR results have been obtained after being averaged over 20 runs by changing the initialisation point of the error patterns. The UDP transfer protocol is chosen. The PER after the MAC is then the PER seen by the application. Emphasis is generally given to a broadcast environment where no MAC retransmission is allowed. The case of a limited number of ARQs is also highlighted for time-bounded live video applications. Moreover, transmissions to over several handhled devices are highlighted, where the channel would be shared, restricting therefore the video bit rates.

8.2.3 Different Packet Sizes

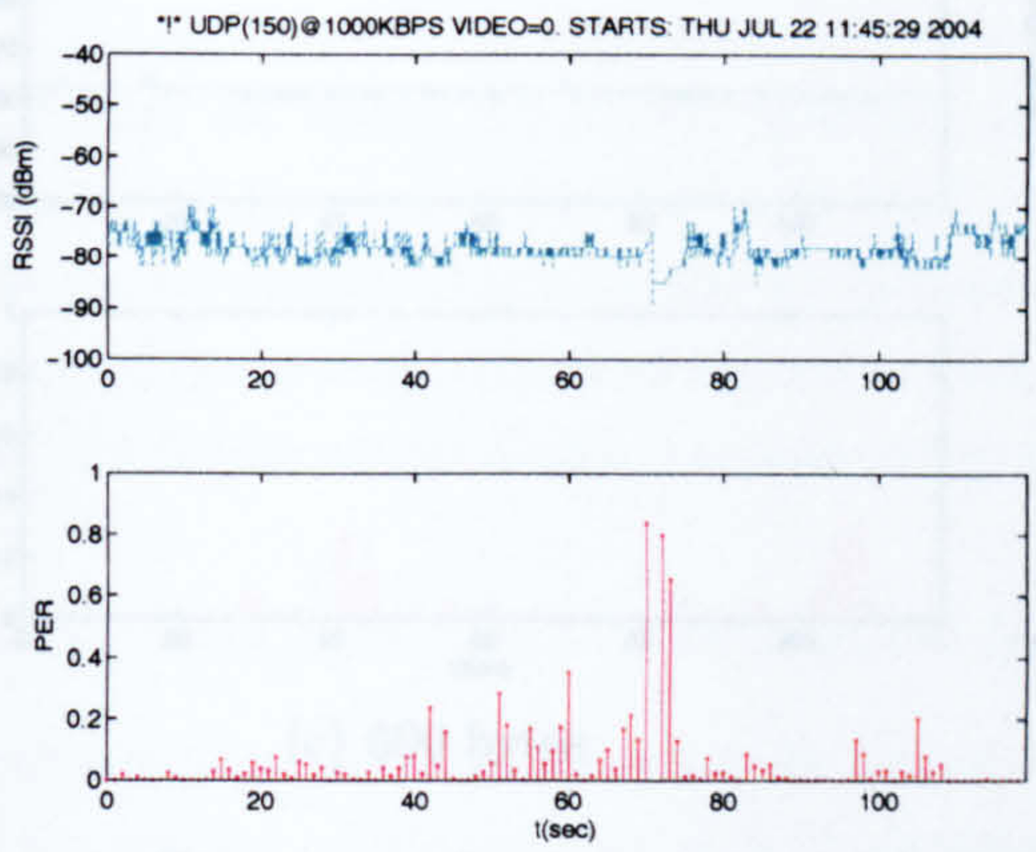
The maximum number of retries at the MAC layer is set to two and the effect of the packet length on the transmission performance are studied. As explained in section

6.6.5, for a given bit rate, the smaller the slice, the more slices are present in a frame and the larger the headers . The different slice headers are presented in table 8.1.

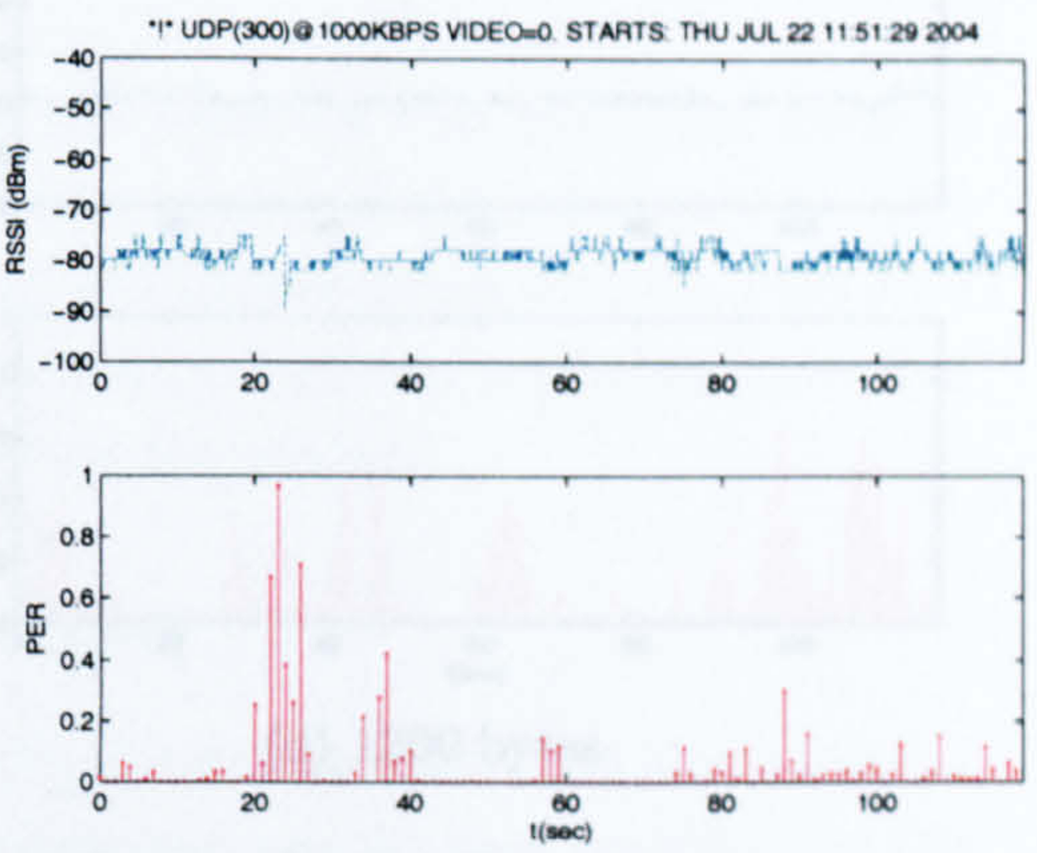
Table 8.1: Percentage Slice Header - 1000kbits/s

Slice Size	*Percentage Slice header*
150 bytes	4.66%
300 bytes	2.27%
600 bytes	1.13%
1200 bytes	0.59%

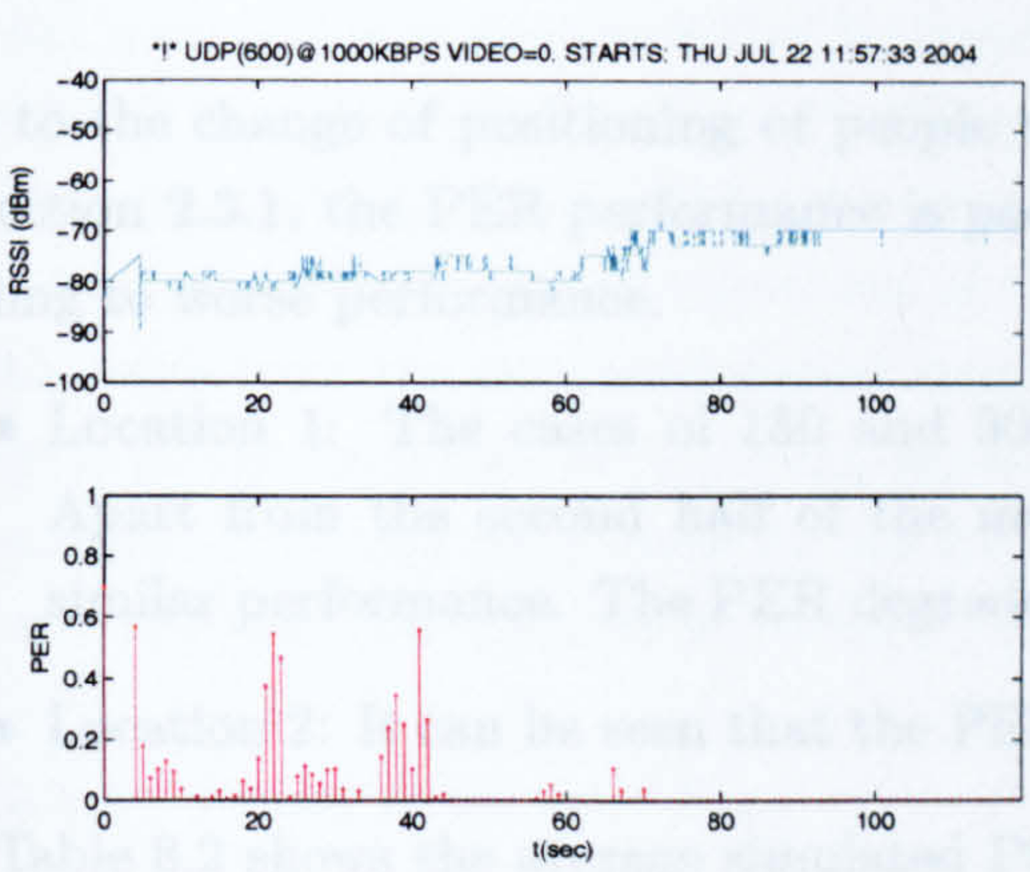
Figures 8.4 and 8.5 show the PER and RSSI performances for the different packet sizes, two different locations respectively within the office.



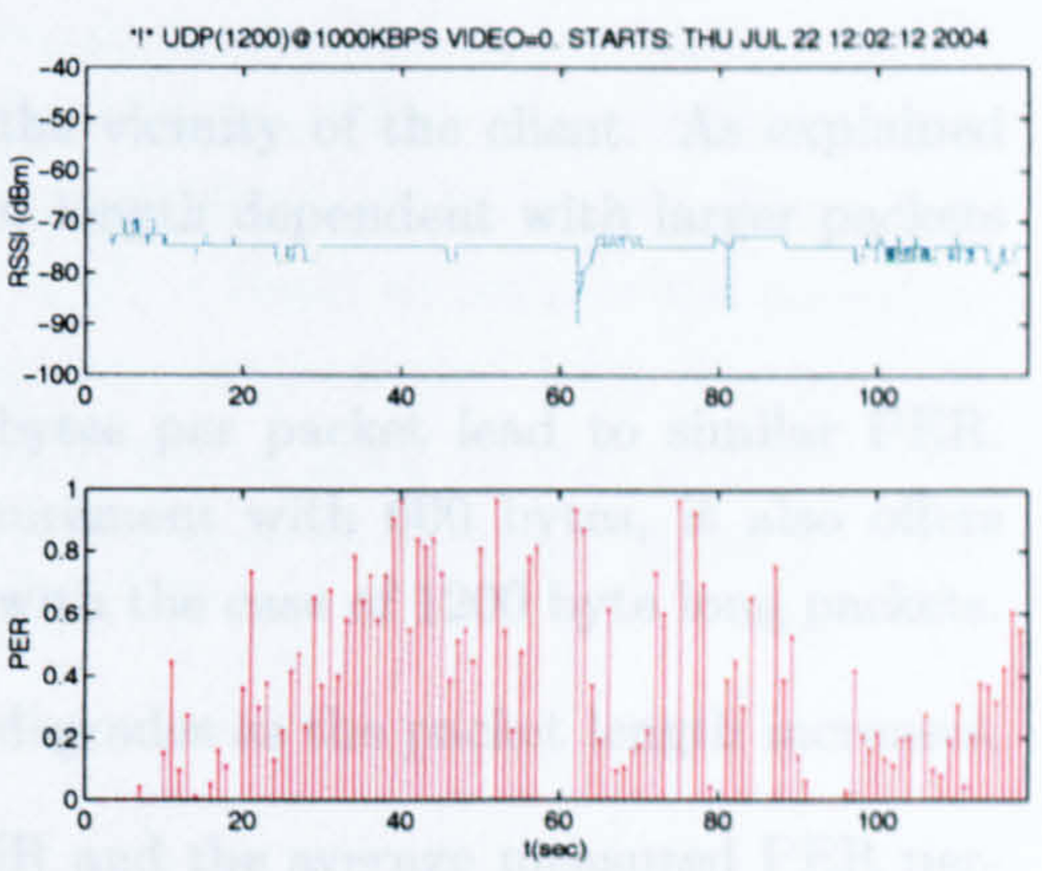
(a) 150 bytes



(b) 300 bytes



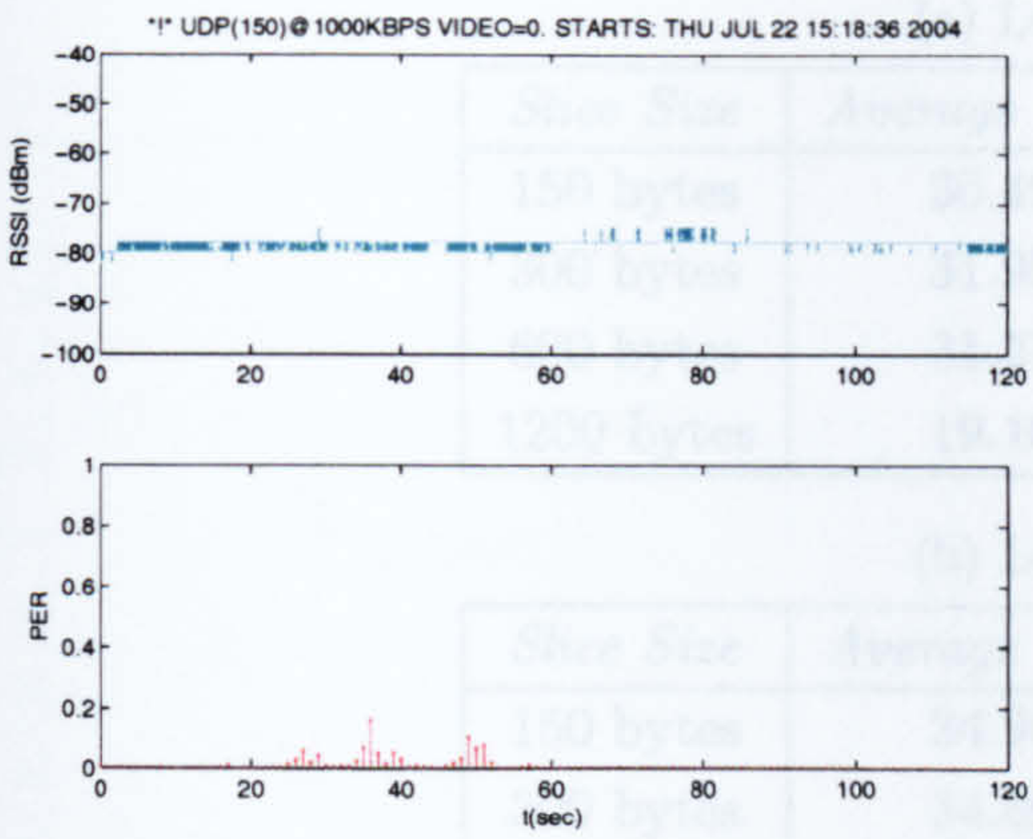
(c) 600 bytes



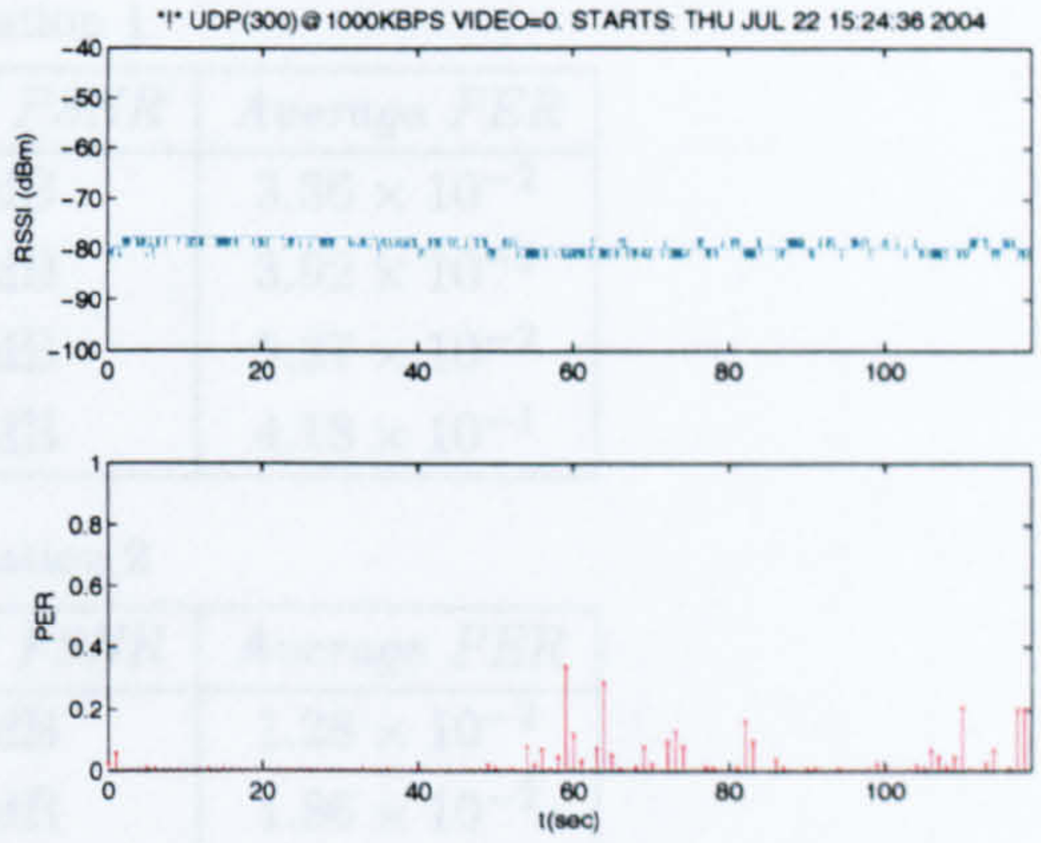
(d) 1200 bytes

Figure 8.4: Performances with Different Packet Sizes - Location 1

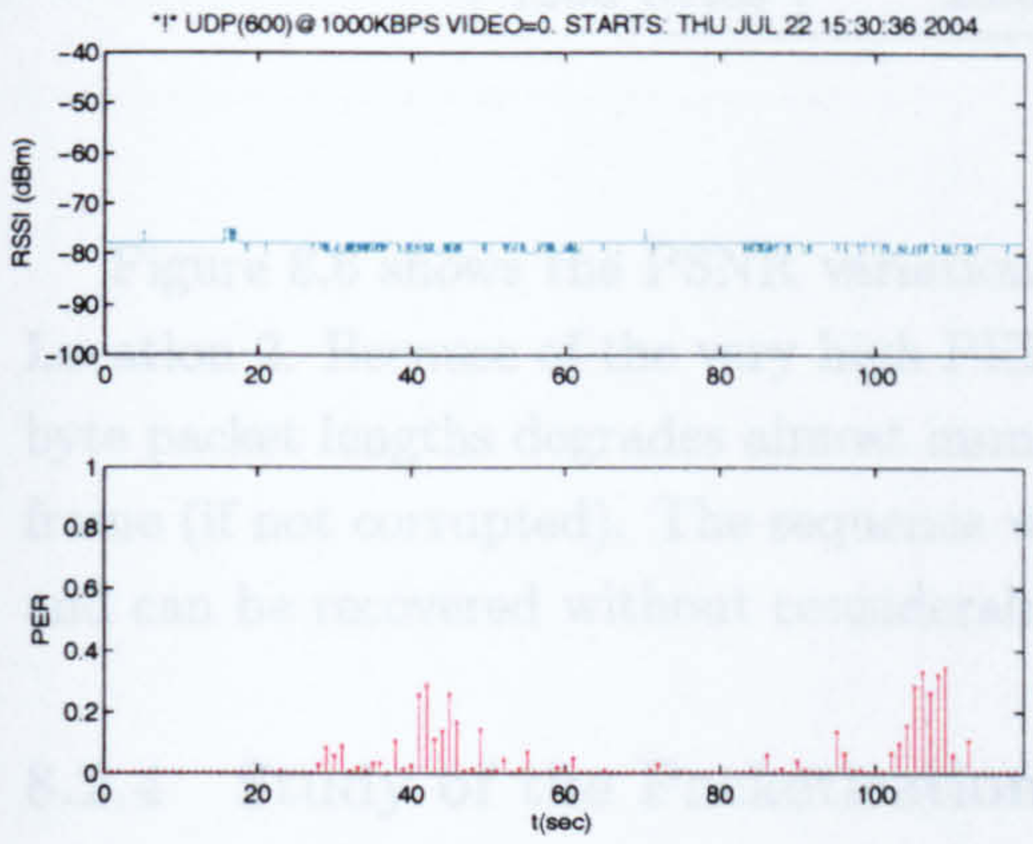
It can be seen that for both locations, the RSSI is almost constant for the different packet sizes, at around -75 dBm for location 1 and -80 dBm for location 2. The case of 600 bytes per packet at location 1 however offers a better RSSI performance (-70 dBm) during the second half of the measurements, and therefore a PER of 0. This is mainly



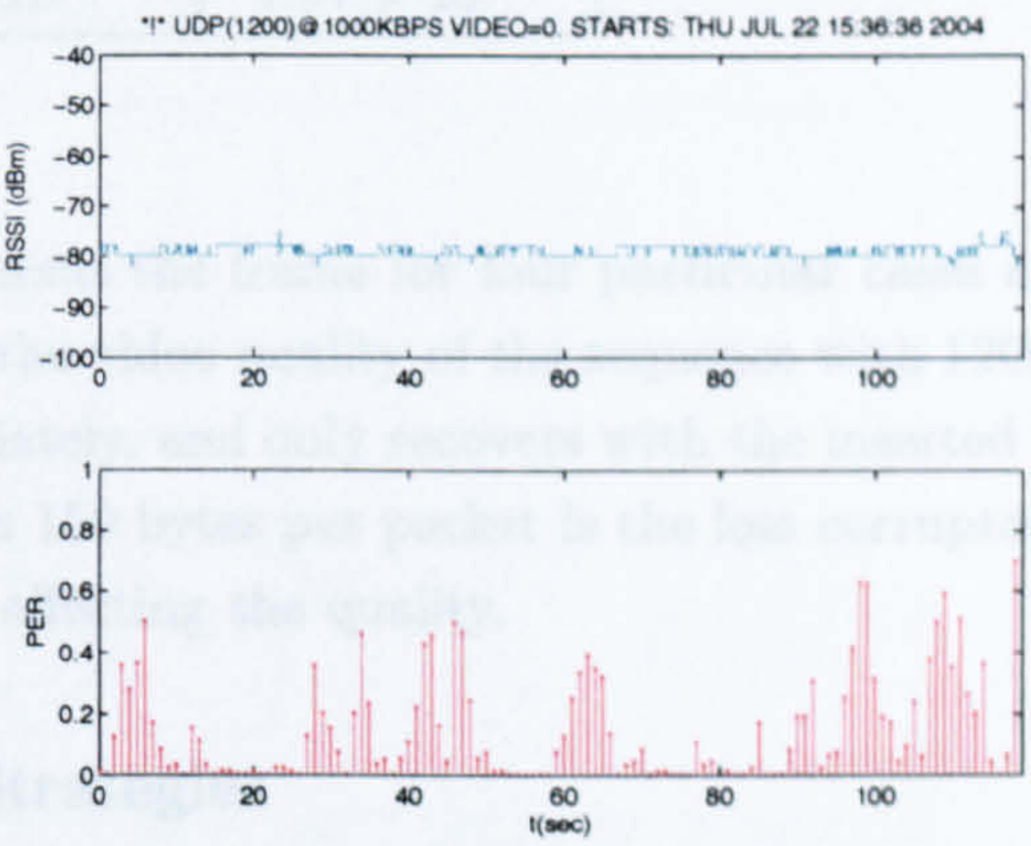
(a) 150 bytes



(b) 300 bytes



(c) 600 bytes



(d) 1200 bytes

Figure 8.5: Performances with Different Packet Sizes - Location 2

due to the change of positioning of people in the vicinity of the client. As explained in section 2.3.1, the PER performance is packet length dependent with larger packets leading to worse performance.

- Location 1: The cases of 150 and 300 bytes per packet lead to similar PER. Apart from the second half of the measurement with 600 bytes, it also offers similar performance. The PER degrades with the case of 1200 byte long packets.

- Location 2: It can be seen that the PER degrades as the packet length increases.

Table 8.2 shows the average simulated PSNR and the average measured PER performance over 20 simulations for both locations. It can be seen that packet sizes of 150 and 300 bytes lead to similar performance for both case. Due to the second error-free transmission part, the packet size of 600 bytes at location 1 offers a similar PSNR performance to 150 and 600 bytes. However, when 1200 bytes per slice are used, PHY packets are larger and are more likely to be corrupted. Therefore the PER degrades, as does the PSNR. In this case, slices contain more MBs, so when a slice is missing more MBs need to be concealed. These remarks are more obvious for Location 2, where the PER and the PSNR degrades as the packet size increases.

Table 8.2: Video Performance for Different Packet Sizes

(a) Location 1

Slice Size	*Average Y PSNR*	*Average PER*
150 bytes	30.49 dB	3.36×10^{-2}
300 bytes	31.95 dB	3.92×10^{-2}
600 bytes	31.27 dB	5.37×10^{-2}
1200 bytes	19.16 dB	4.18×10^{-1}

(b) Location 2

Slice Size	*Average Y PSNR*	*Average PER*
150 bytes	34.90 dB	1.28×10^{-2}
300 bytes	34.09 dB	1.86×10^{-2}
600 bytes	31.86 dB	3.69×10^{-2}
1200 bytes	26.62 dB	1.07×10^{-1}

Figure 8.6 shows the PSNR variation versus the frame for four particular cases at Location 2. Because of the very high PER, the video quality of the sequence with 1200 byte packet lengths degrades almost immediately, and only recovers with the inserted I frame (if not corrupted). The sequence with 150 bytes per packet is the less corrupted and can be recovered without considerably affecting the quality.

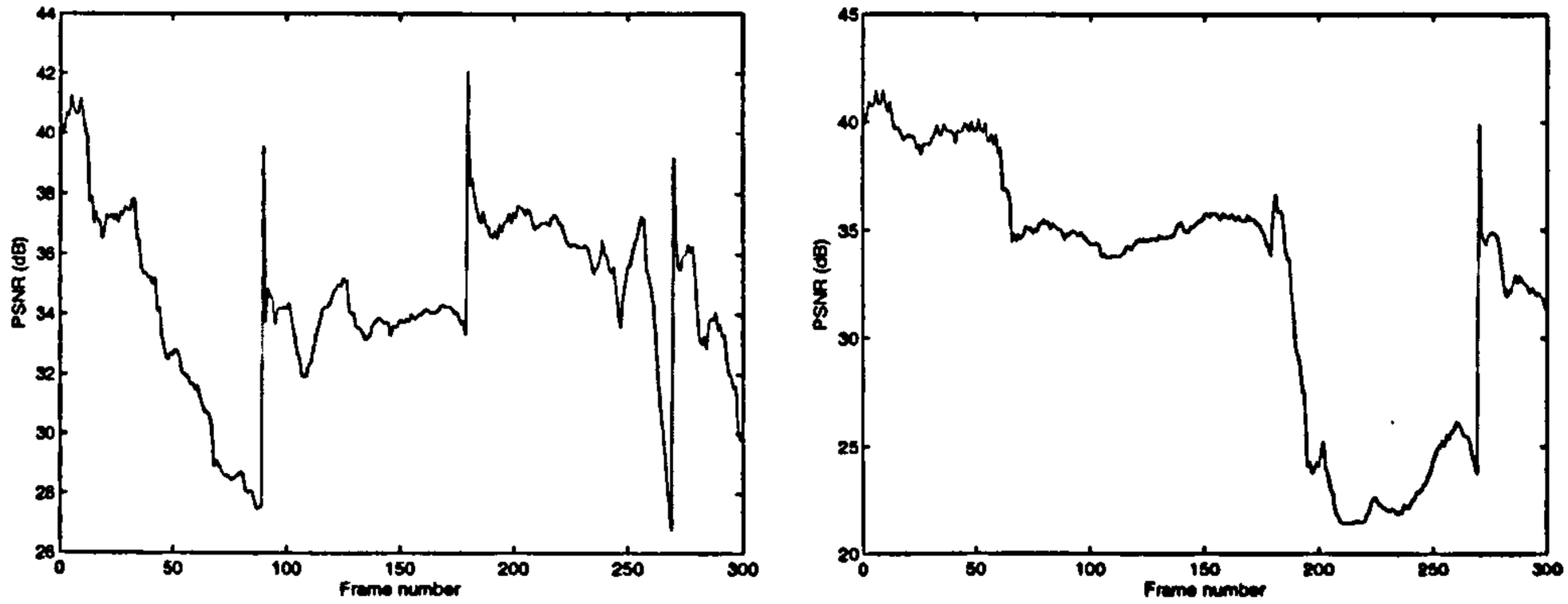
8.2.4 Study of the Packetisation Strategies

In this section, the packetisation developed in chapter 6 is studied. The study focuses on the impact of the number of NAL units mapped onto one wireless packet when the NAL units size is fixed and when the wireless packets have a fixed size is investigated. It has been shown in chapter 6 that by increasing the packet size on the wireless MAC, the MAC overhead is reduced and the throughput increased. One suggested solution is to map/encapsulate several NAL units onto one IEEE 802.11 MAC packet. Because of the lack of flexibility, the modifications of the MAC in order to retrieve the error-free NAL units from one corrupted packet have not been implemented here. Instead, the impact of the encapsulation with the legacy IEEE 802.11 MAC on the decoded video is studied. For these measurements, the maximum number of retries at the MAC is set to two and the UDP transfer protocol is chosen.

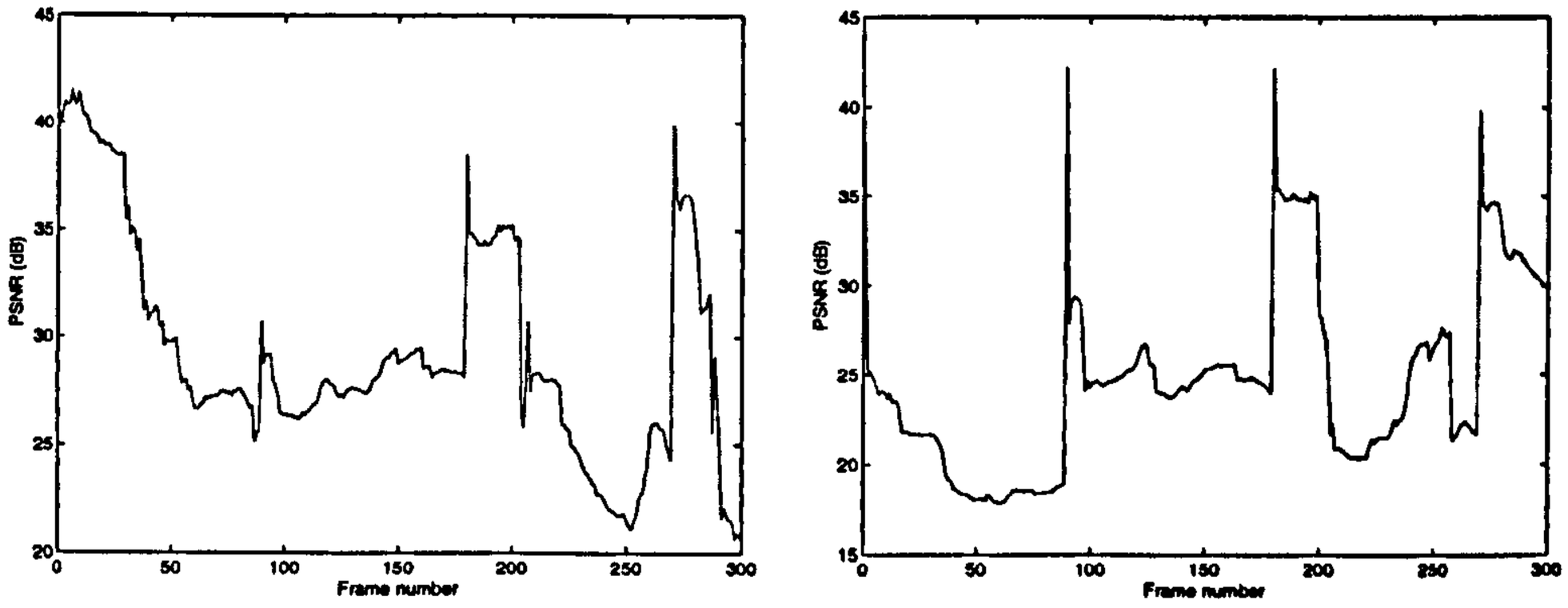
8.2.4.1 Fixed NAL unit Size

In this section, the video is encoded with a fixed NAL unit length of 150 bytes. One, two, four and eight NAL units are mapped to one MAC frame/PHY packet of size 150, 300, 600 and 1200 bytes respectively. Figure 8.5 shows the PER performances used for the simulations.

Table 8.3 shows the average simulated PSNR and average measured PER over 20 runs and highlights the need for a system to retrieve possible error-free NAL units



(a) 150 bytes, PSNR=34.5dB, PER=1.1×10^{-2} (b) 300 bytes, PSNR=32.7dB,PER=2.4×10^{-2}



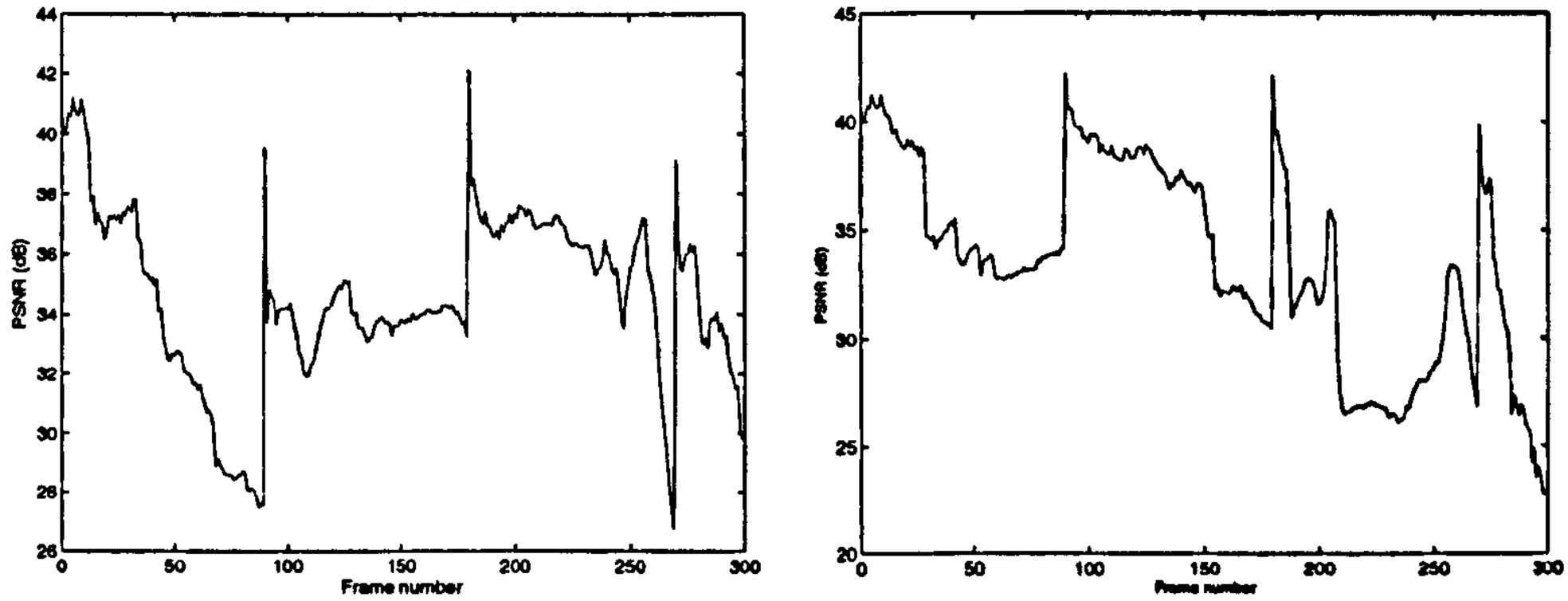
(c) 600 bytes, PSNR=29.3dB, PER=3.3×10^{-2} (d) 1200 bytes, PSNR=23dB, PER=1.4×10^{-1}

Figure 8.6: PSNR for Packet Size Comparison - Location 2

Table 8.3: Video Performance for Different Packet Mapping with Fixed NAL size

Number of NAL Units per Packet	*Average PSNR*	*Average PER*
1	34.90 dB	1.28×10^{-2}
2	34.30 dB	1.92×10^{-2}
4	33.11 dB	2.71×10^{-2}
8	26.61 dB	1.23×10^{-1}

within one wireless packet. Larger packet lengths clearly increase the throughput but also are more likely to be corrupted. Moreover, more video data are lost when one packet containing encapsulated NAL units is lost. Concatenating NAL units at the RTP layer with the aggregation mechanism does not however provide any recovery system within the RTP packet. Figure 8.7 confirms for a particular run the fact that multiple NAL units per packet without any recovery mechanism degrades the quality of the received video when fixed NAL unit sizes are used.



(a) 1 NALU, PSNR= 34.5dB, PER=1.1×10^{-2} (b) 2 NALUs, PSNR=33.6dB, PER=3.1×10^{-2}

(c) 4 NALUs, PSNR=32.9dB, PER=2.3×10^{-2} (d) 8 NALUs, PSNR=27.2dB, PER=1.2×10^{-1}

Figure 8.7: PSNR for Encapsulation Comparison - Fixed NAL Size

8.2.4.2 Fixed Wireless Packet Size

In this section, the wireless packet has a fixed size of 1200 bytes. One, two, four and eight NAL units of size 1200, 600, 300 and 150 bytes respectively are mapped to one packet. Because video sequences have different NAL unit sizes for a similar bit rate, the slice overhead influences the quality. Figure 8.5(d) shows the PER performances used for the simulations. Table 8.4 shows the average simulated PSNR and average measured PER performances. Similar PSNR values are achieved for different mappings. Recovering error-free NAL units, if there are any, will then improve the video quality.

Table 8.4: Video Performance for Different Packet Mapping with Fixed Wireless Packet Size

Number of NAL Units per Packet	*Average PSNR*	*Average PER*
1	26.18 dB	1.19×10^{-1}
2	25.94 dB	1.20×10^{-1}
4	26.31 dB	1.21×10^{-1}
8	26.41 dB	1.20×10^{-1}

8.2.5 Study of the Video Quality based Link Adaptation algorithm

8.2.5.1 Description

In this section, the throughput-based and the studied video quality-based link adaptation algorithms investigated in chapter 7 are compared using real measurement data. The IEEE 802.11g $BELKIN^{TM}$ card is mounted on the client. The server, mounted with the IEEE 802.11g $Buffalo^{TM}$ card, is kept static while the client is moved away at walking pace in an outdoor LOS environment. The proposed link adaptation algorithm based on video quality could not be implemented in the chipset because of a lack of flexibility in the cards. By using the log files, error patterns have been generated for each mode with their respective packet size and have been used in order to simulate both algorithms. A 39s video sequence has been encoded with H.264 at 30Hz using parameters taken from Set 2 (see table 8.5). Moreover, the first frame of each encoded video is an IDR picture, meaning that it is INTRA coded, but also that the reference picture buffer is emptied. In this way no error can propagate over the IDR picture. This IDR picture has been ensured to be received error-free with a TCP link. Note that mode 2 and mode 4 have not been used since it appeared from chapter 7 that they are not operating in the algorithm.

Table 8.5: Video Parameters - Set 2

Mode	*Video Bit Rate (kbits/s)*	*Packet Length*
1	500	188 bytes
3	1000	376 bytes
5	2000	752 bytes
6	3000	1128 bytes
7	4500	1692 bytes

A unicast link with no ARQ has been used. Using PER thresholds as described in section 7.3.3.3 is unfortunately not practical in the case of these particular measurements since accuracy at such low PER could not be obtained. Instead, a simpler solution with C/N thresholds has been considered using figure 7.16(a). The C/N thresholds are given in table 8.6. C/N has been derived from the RSSI using equation 8.1

[46].

$$C/N(dB) = RSSI(dBm) - (K.T.B(dBm) + NF(dB)) \tag{8.1}$$

where K is Boltzmann's constant, T is the temperature (290 °K) and B is the bandwidth. NF is the noise figure (8dB) and $K.T.B(dBm) + NF(dB)$ is equal to -92 dBm [46].

Table 8.6: Mode Selection Based on C/N
(a) Video Quality Algorithm (b) Throughput Algorithm

C/N in dB		
From	*To*	*Mode*
-	36	7
36	29.5	6
29.5	22	5
22	15	3
15	-	1

C/N in dB		
From	*To*	*Mode*
-	23	7
23	16	6
16	10.5	5
10.5	2.5	3
2.5	-	1

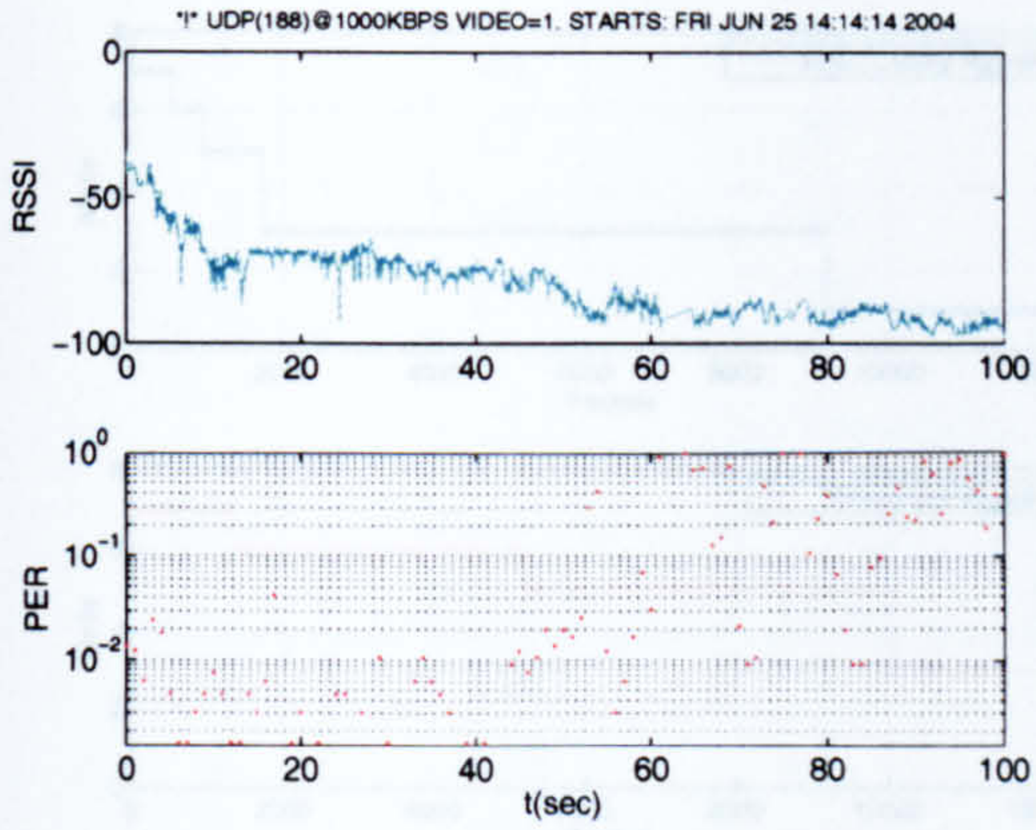
8.2.5.2 Results

Figure 8.8 shows the PER and RSSI for each mode along the route. The gaps in the PER graphs are due to a loss in connection. This could be attributed to pedestrians walking closely in front of the server antennas, losing therefore the LOS configuration. These losses also occur when the client reaches the limit of the coverage area of the server. It explains therefore the shorter logging duration for mode 7.

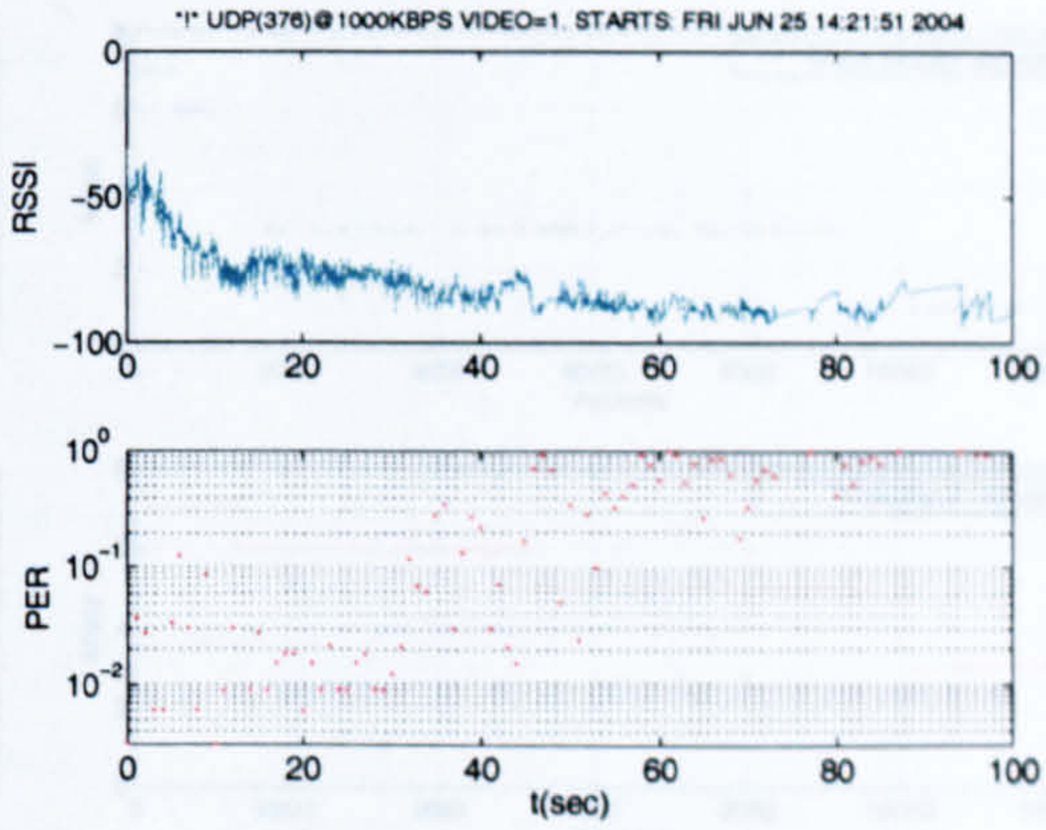
It can be seen that as the client moves away from the server, the RSSI decreases. The PER is then naturally increasing. The PER for mode 1 is however increasing later (at around 50s) compared to mode 3 (at around 30s) and mode 5 (at around 40s). Mode 6 presents relatively bad PER performance, even when the client stands close to the server. This could be explained by the possible positioning of people in the vicinity of the server. Moreover, PER performance for Mode 7 decreases and drops rapidly, illustrating the small coverage range of the 3/4 rate 64 QAM mode.

Using table 8.6 and figure 8.8, it is possible to determine the mode that one packet should be transmitted on, for both algorithms, as shown in figure 8.9. This figure also shows the missing packets. The video quality based algorithm switches down from one mode to a lower one quicker than the throughput-based algorithm. Figure 8.10 shows how the mode changes in time. It also shows which mode is used on a frame basis. This data is summarised in table 8.7.

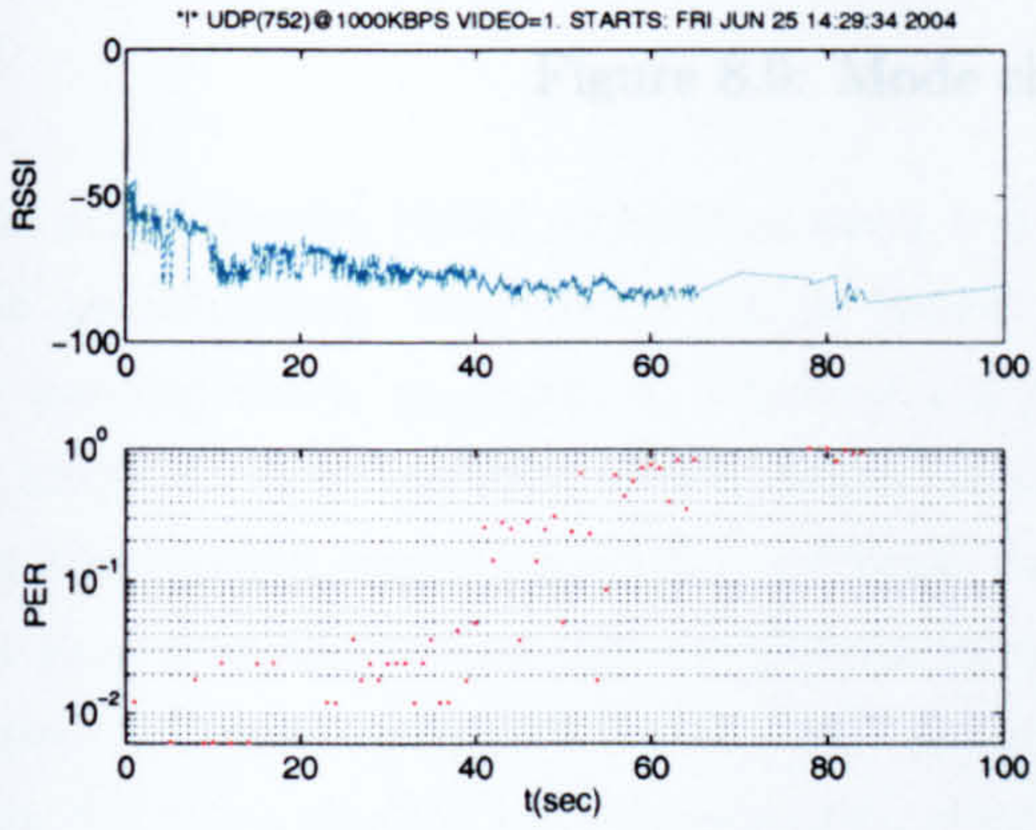
Figure 8.11 shows the PER and received PSNR for both algorithms. The PER and the lost packets of figure 8.9(b) correspond to the error patterns applied to the encoded sequence. The video quality based algorithm switches from mode 7 to mode 6 and from mode 6 to mode 5 quickly compared to the throughput based algorithm.



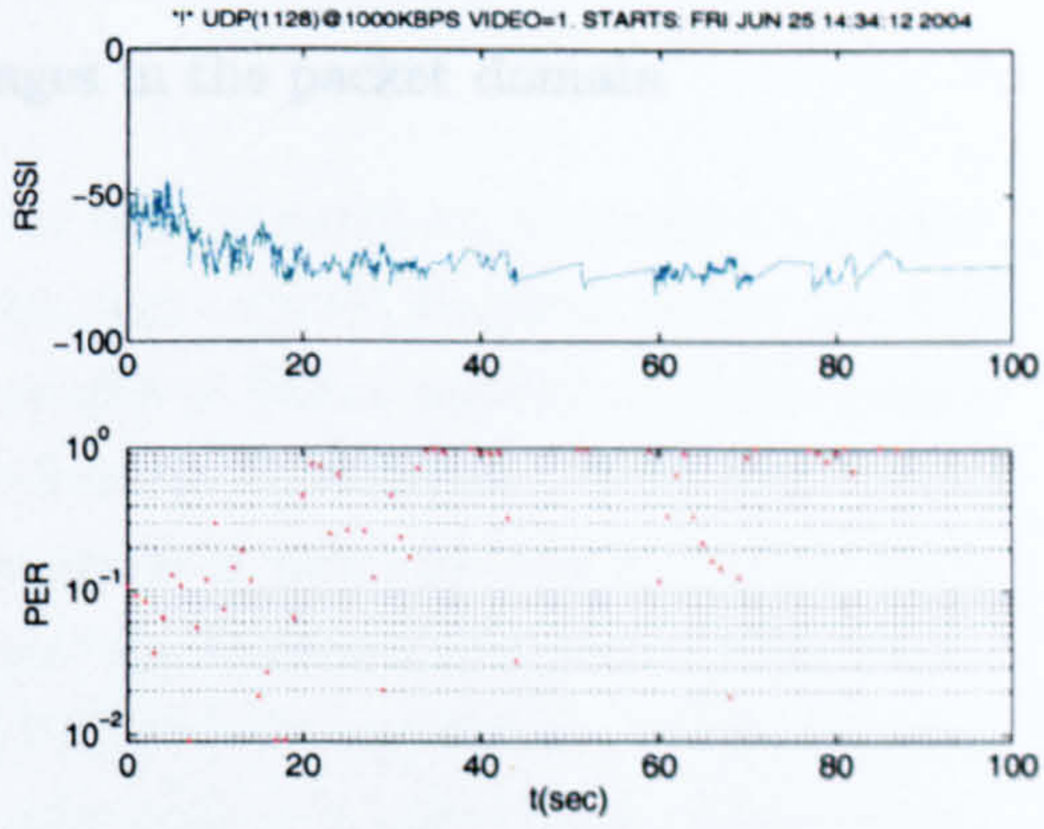
(a) Mode 1



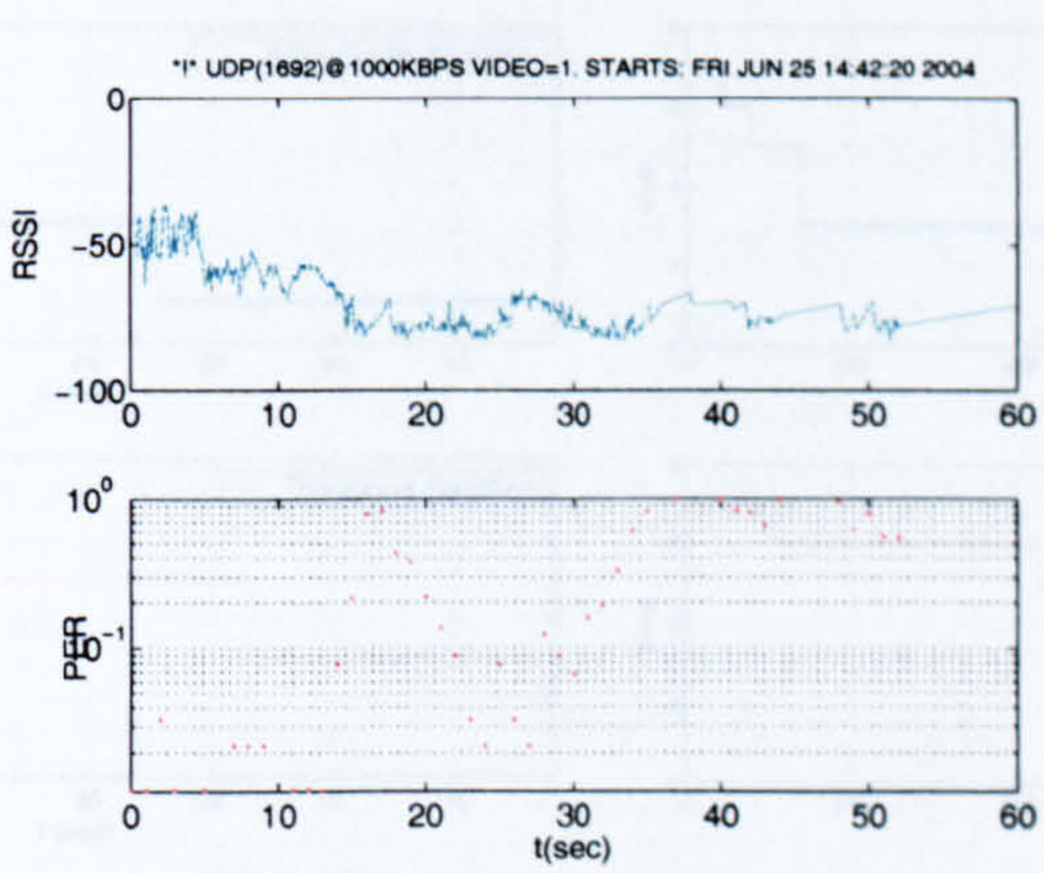
(b) Mode 3



(c) Mode 5

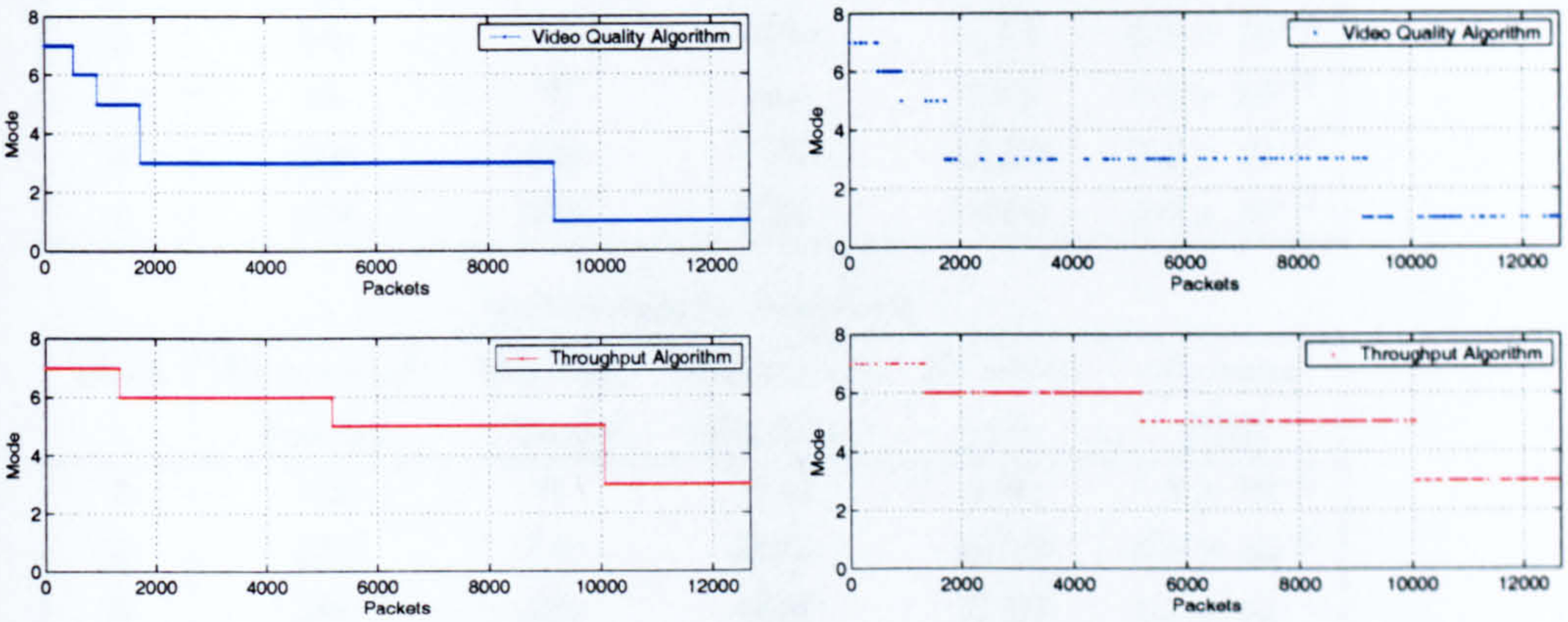


(d) Mode 6



(e) Mode 7

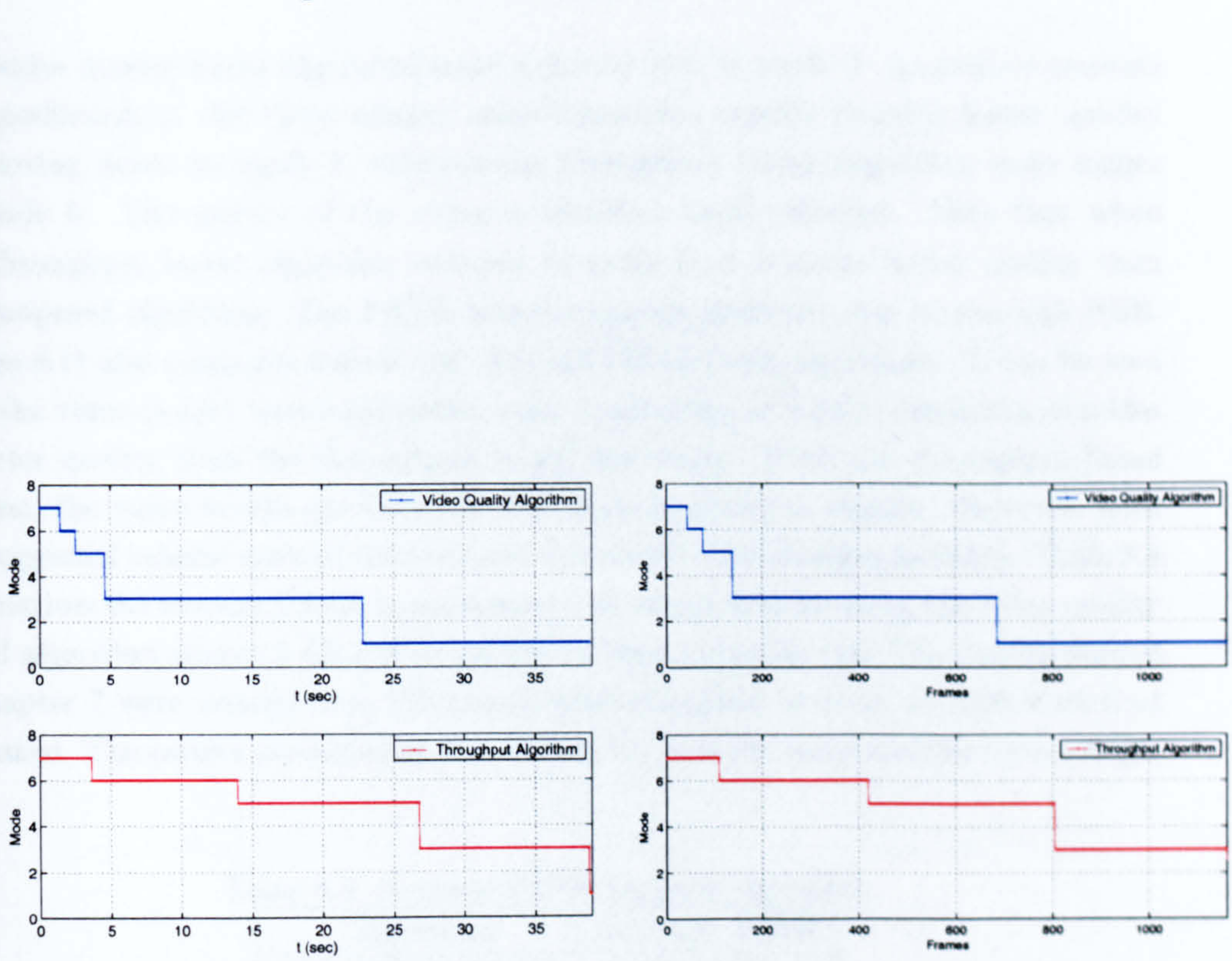
Figure 8.8: PER and RSSI along the route



(a) Transmitted Packets

(b) Missing Packets

Figure 8.9: Mode changes in the packet domain



(a) Mode versus Time

(b) Mode versus Frame

Figure 8.10: Mode changes in the time domain

Table 8.7: Frame/Mode Repartition

(a) Video Quality Algorithm

Mode	*Number of Frames*	*Starting Frames*	*Number of Packets*	*Duration in s*	*Average PER*
7	43	0	526	1.43	1.4×10^{-2}
6	34	43	428	1.13	9.2×10^{-2}
5	61	77	784	2.03	5.4×10^{-3}
3	548	138	7437	18.26	2.2×10^{-2}
1	479	686	7721	15.96	9.9×10^{-3}

(b) Throughput Algorithm

Mode	*Number of Frames*	*Starting Frames*	*Number of Packets*	*Duration in s*	*Average PER*
7	110	0	1342	3.66	1.3×10^{-2}
6	310	110	3843	10.33	4.3×10^{-1}
5	385	420	4866	12.83	1.2×10^{-1}
3	360	805	4860	12.00	1.9×10^{-1}

The video quality based algorithm stays a shorter time in mode 6. As mode 6 presents bad performance, the video quality based algorithm rapidly re-gains better quality by moving down to mode 5, whereas the throughput based algorithm stays longer in mode 6. The quality of the video is therefore badly affected. Note that when the throughput based algorithm switches to mode 5, it presents better quality than the proposed algorithm. The PSNR however quickly decreases due to the high PER. Figure 8.11 also compares frames 100, 400 and 700 for both algorithms. It can be seen that the video quality based algorithm, even if operating at a lower resolution provides a better quality than the throughput based algorithm. With the throughput-based scheme, the video breaks up whenever the mode is about to change. However, with the suggested scheme smooth transmission can occur, thus enabling mobility. Table 8.8 summarises the average PSNR improvement and shows that by using the video quality based algorithm almost 5 dB can be gained on this particular run. The results showed in chapter 7 were average over 100 transmitted sequences in order to have statistical relevance. The results presented in this section are however computed for only a single run.

Table 8.8: Average PSNR for both algorithm

Algorithm	*Average PSNR*
Throughput based	23.35 dB
Video Quality based	28.66 dB

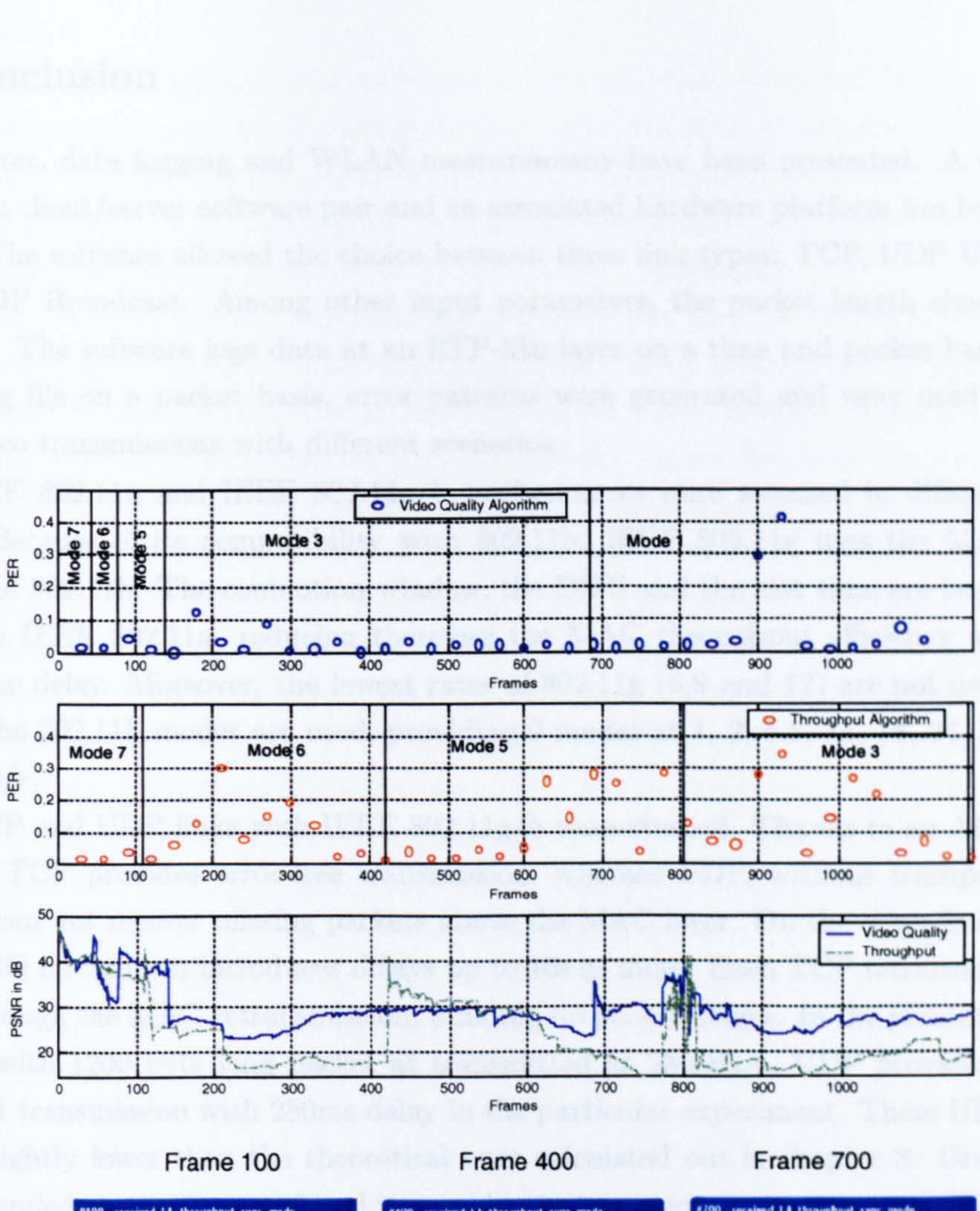


Figure 8.11: PSNR versus Frame for Link Adaptation Comparison

8.3 Conclusion

In this chapter, data logging and WLAN measurements have been presented. A description of a client/server software pair and an associated hardware platform has been presented. The software allowed the choice between three link types: TCP, UDP Unicast and UDP Broadcast. Among other input parameters, the packet length should be specified. The software logs data at an RTP-like layer on a time and packet basis. From the log file on a packet basis, error patterns were generated and were used to simulate video transmissions with different scenarios.

The IEEE 802.11a and IEEE 802.11g/b performances were assessed in different scenarios. Because of its compatibility with 802.11b, IEEE 802.11g uses the MAC parameters of 802.11b. The contention window, the DIFS and the slot time are larger compared to IEEE 802.11a, reducing therefore the MAC throughput efficiency and increasing the delay. Moreover, the lowest rates of 802.11g (6,9 and 12) are not used, but rather the 802.11b modes are used, providing 9 modes at 1, 2, 5.5, 11, 18, 24, 36, 48 ,54 Mbits/s.

Both TCP and UDP links with IEEE 802.11g/b were studied. Thanks to an ARQ mechanism, TCP provides error-free transmission, whereas UDP, without transport layer ARQ, can not recover missing packets above the MAC layer. On the other hand, the TCP ARQ mechanism introduces delays up to 10s or more. Each TCP retransmission goes through the MAC retransmission, building up packet delays. In the presented experiment with 1200 byte long packet at transmitted at 2Mbits/s, UDP provides a delay-limited transmission with 250ms delay in the particular experiment. These UDP delays are slightly lower than the theoretical ones calculated out in chapter 3. Given the time-bounded requirements of real-time video transmission, it is clear that UDP should be implemented

The need and use of flow control for the transmission rate has been highlighted, especially for transmission with a UDP link. To avoid lost packets at the transmitter when the channel conditions are bad, the transmitter should be able to buffer packets and control the flow. A buffer at the UDP level is not possible. UDP does not provide enough flexibility to control the transmission rate and the application can not probe the UDP buffer to check whether the buffer is ready to accept another packet or not. This is hence presented as a weakness of the UDP protocol. One way to control this transmission rate is that the application should have a low level MAC access in order to check the status of the MAC buffer and therefore adapt its flow. At the receiver, a buffer is needed in order to correct jitter delay at the decoder.

Among several parameters that can be tuned for a real-time video transmission, the influence of packet size has been studied in terms of PER. The larger the packet, the more likely it is to be corrupted and the more penalised the video quality. The packetisation study of chapter 6 has been investigated. The modifications to the MAC could not be implemented but the importance of a recovery scheme has been outlined when several NAL units are encapsulated into one packet for transmission. Concatenating NAL units allows better throughput efficiency with smaller overheads. However,

larger packets are more likely to be corrupted, and within these corrupted packets, some NAL units may be error-free. If recovered, these error-free NAL units can enhance the video quality. The way NAL units are mapped onto the underlying network is critical and is not specified by any standards. Simulation results have been carried out in an environment where limited numbers of ARQs is a key requirement.

Finally, the study of Link Adaption algorithms as described in chapter 7 has also been discussed in this chapter. The studied algorithm based on video quality could not be implemented because of a lack of flexibility in the card. Instead of using a PER threshold as recommended in chapter 7, a much simpler method has been adopted using C/N thresholds. Using the error pattern for each mode on a route in a LOS, the proposed rate and video adaptation algorithm switched down to lower modes earlier than the throughput based approach. It was seen that with the proposed algorithm the video quality remained better than the other algorithm, where the video quality degraded too much before switching down as the mobile moves away. With the throughput-based scheme, the video breaks up whenever the mode is about to change. However, with the suggested scheme smooth transmission can occur, thus enabling mobility. A 5dB gain was shown. Using a more accurate technique with PER thresholds as recommended should improve further the performance. The link adaptation algorithm is therefore a critical issue that needs to be addressed by manufacturers. To have good, reliable and efficient transmission, the algorithm needs to be adapted to the channel conditions, but more importantly, to the requirements and types of application. This is especially true for real-time video transmission with a limited number of ARQs.

Chapter 9

Conclusions and Further Work

This thesis has concentrated on the problem of multimedia transmission over COFDM-based Wireless LANs. Emphasis has been placed the specific cases of the PHY/MAC layers of IEEE 802.11a at 5.2GHz, IEEE 802.11g at 2.4GHz and the H.264 video encoder. A cross layer approach, where layers interact, coordinate their efforts and share information, was proposed. After a study of the COFDM-based physical layer and the IEEE 802.11 MAC, the H.264 standard was presented. In order to reduce the use of ARQ for MPEG-2 system-like transmissions, enhancements to the PHY layer were proposed, using combined STBC and FEC. Packetisation strategies for mapping several NAL units into one MAC frame were devised. A new link adaptation algorithm was also presented, optimising the video quality rather than the error-free data throughput. Finally, an investigation and characterisation of performance using practical data measurements was reported.

9.1 Summary

The COFDM-based PHY layer of IEEE 802.11a/g was studied in chapter 2. Performance results were presented in terms of BER, PER and throughput. IEEE 802.11a/g allows variable length PDUs (up to 4096 bytes) to be sent, whereas Hiperlan /2 uses a fixed PDU size of 54 bytes. The BPSK 1/2 rate mode (mode 1) provides the most reliable transmission mode, at the expense of the lowest nominal bit rate, whereas the 64 QAM 3/4 rate mode (mode 7) offers an unreliable transmission but with the highest nominal bit rate. The throughput is PER and mode dependent. The higher the PER, the lower the throughput. PER performance is also packet size dependent for IEEE 802.11a/g. Larger packets are more likely to be corrupted than smaller ones and therefore results in a lower link throughput performance. The operating mode and the size of the transmit packets are critical for system performance.

In chapter 3, the asynchronous IEEE 802.11 Medium Access Control based on CSMA/CA was studied. Emphasis was given to the mandatory DCF mode with its basic access and RTS/CTS schemes. The stop and wait ARQ is a mandatory feature used to retransmit corrupted packets. A random back-off mechanism is used in order to avoid collision. Throughput and delay studies were presented. The throughput is

packet length dependent and the IEEE 802.11 MAC was shown to have a poor overhead efficiency. Long packets offer a better throughput than small packets because of the reduced overheads. With mode 1, 500 byte long packets have a 25% overhead, whereas 1500 byte long packets only have a 10% overhead. Long packets at the MAC are therefore preferable for high bit rate applications, such as video applications. When combined with the PER, the throughput is further reduced. The ARQ mechanism introduces packet delays that increase with the number of retransmissions allowed. The trade-off between PER, delay and throughput has been highlighted. The maximum number of retries is not in the specification and is left to the manufacturers. Time bounded applications should limit the number of ARQs since a packet would suffer a 20 ms delay if it is required to be retransmitted 8 times with the RTS/CTS scheme (assuming the IEEE 802.11a MAC parameters). The particular case of broadcast transmission does not allow retransmission. Because of its backward compatiblity with IEEE 802.11b, IEEE 802.11g has larger overheads, smaller throughput and larger delays than IEEE 802.11a. By using a Markov Chain analysis with DCF, it was shown that as the number of users increases, the RTS/CTS scheme provides better throughput performance than the basic scheme since collisions occur on the RTS frame rather than on the data frame. However, with small packets, the basic access scheme has a better use of channel resources and offers a better throughput. The optional PCF access mode designed for time bounded applications was presented and the lack of QoS in the legacy IEEE 802.11 MAC support was highlighted.

The basics of video coding were reviewed in chapter 4. The coding features of the MPEG-2 video standard were presented. MPEG-2 also incorporates a System Layer, defining the packetisation from MPEG-2 access units to transport stream, as well as the multiplexing with audio streams. Error resilience and concealment techniques have been presented for video transmission over error-prone channels. These techniques include the partitioning of frames into smaller entities (frames, slices or group of blocks). The new H.264 video standard has been presented with its new coding features. Its Video Coding Layer (VCL) has improved the compression efficiency with the combination of various techniques. H.264 has been designed so that it can be easily transmitted over many types of system using its Network Abstraction Layer (NAL), providing an interface with underlying networks. Emphasis has been given to the error-resilience tools developed in the standards, including FMO, multiple reference pictures, slice and slice group structures. H.264 compression efficiency out performs the previous MPEG2 and H.263 standards, and due to its error robustness features and its network abstraction flexibility, H.264 is adapted to provide reliable transmission over error-prone channels such as those presented by WLANs.

Possible enhancements for MPEG-2 video transmission over the IEEE 802.11a PHY layer were studied in chapter 5. Real-time video transmission cannot bear too many retransmissions and ARQ is a mandatory feature of the IEEE802.11 MAC layer. In order to ensure MPEG-2 delivery without relying on the use of the MAC ARQ, the PER at the PHY layer needs to be reduced. Similarly to DVB-T and DVB-S, the use of Reed

Solomon Codes with the COFDM-based IEEE 802.11a PHY has been investigated. However, packets can get highly corrupted, the PHY PER is improved in a small proportion, and the BER is almost unchanged. To reduce heavily corrupted packets, Space Time Block Codes (STBC) providing spatial diversity have been introduced. STBC uses multiple transmit and receive antennas and is simple and attractive. PER and BER performance as well as byte and bit in error distributions showed that a 8dB gain is achievable for mode 3 with a 2Tx-2Rx case for a PER of 10^{-2}. STBC reduces the density of errors allowing the use of an RS code. Combinations of STBC and RS codes correcting up to 8 and 32 bytes have been studied. Results showed that the use of layer 2 ARQ is considerable reduced. At the expense of a lower error-free transmission rate due to RS overheads, PHY PER performances are improved, thus minimising the number of MAC ARQ. However, under bad channel conditions, the proposed scheme strongly improved the throughput since ARQ is reduced and less bandwidth is consumed for retransmission. 10dB gains were observed for a PER of 10^{-2} with a 32 byte correction code and a 2Tx-2Rx STBC case. STBC combined with RS is therefore a possible robust PHY layer enhancement for future multimedia transmission systems based on WLAN.

The low throughput efficiency of the IEEE 802.11 MAC layer, as addressed in chapter 3, and the need for robust video transmission were stressed in chapter 6. Two packetisation strategies were proposed and characterised in order to offer enhanced H.264 video transmission over an IEEE 802.11 based MAC in a UDP/IP environment. The first strategy is performed at the MAC layer, while the second is developed at the application layer. Emphasis has been given to robust broadcast transmissions to several handheld devices sharing the same channel and to low latency unicast transmissions without strong reliance on ARQ. The first proposed strategy provided good throughput for video transmission, as well as guaranteed video robustness. This was achieved via the design of an aggregation scheme at the MAC layer, where several IP packets (containing NAL units) were mapped into one single MAC frame. The robustness of the video was ensured with the use of small video slices (NAL units), and by recovering any error-free IP packets in one MAC frame. The first strategy required the legacy MAC to be modified, including the design of a *Block_Ack* mechanism, an *aggregation field* in the MAC header and the addition of a selective-repeat ARQ at the MAC frame level. Each IP packet was individually 'check summed' and the CRC was applied to the MAC header only. Enhancements to the MAC throughput have been presented, as well as improvements in terms of NAL unit Error Rate (NER) and PSNR for H.264 video transmission. The proposed system behaved as if single NAL units were transmitted one by one, therefore allowing a low NAL unit Error Rate for small NAL units. Multiple NAL units per MAC frame allowed throughput improvements and a 3 dB gain in PSNR at a PER of 10^{-2} with the presented method when 10 NAL units were mapped. Moreover, for a fixed PHY length, a 2.5 dB gain at a PER of 3×10^{-2} was achieved by mapping 8 small NAL units with the proposed method, compared to the legacy MAC case where using a single large NAL unit per MAC frame is used. This first proposed strategy guaranteed MAC efficiency as well as good video quality. The second strategy

presented studied the use of an interleaved cross packet FEC at the application layer with a Reed-Solomon erasure correction code. Different parameters, such as coding rate, interleaving depth and correction capability, were considered and their effects on the received video quality were studied. By using a coding rate of 0.875 with a depth of 8, a 6dB gain in PSNR could be achieved over the standard for a PER of 10^{-2}. A combined scheme using multiple NAL units mapped into one MAC frame with a cross packet FEC is possible and would greatly enhance the system performance.

A link adaptation strategy designed for H.264 video transmission over the IEEE 802.11 MAC and IEEE 802.11a PHY layers was presented and characterised in chapter 7. The proposed approach was motivated by previous algorithms that focused on maximising the error free data throughput without taking into account the nature of the content and the time-bounded requirement of real-time video transmissions.

For low latency video applications, extensive use of retransmission is not possible and the overall received video quality has to be optimised rather than the throughput. The main idea behind this study was that each PHY mode was carrying different video bit rates, and then video qualities. The video encoder therefore modifies its encoding parameters depending on the transmission mode of the PHY. The chosen mode is the one that provides the best video quality. The investigation showed that a natural adaptation based on PSNR was not feasible and a more practical approach was proposed based on PER thresholds. The choice of PER thresholds was shown to be critical. For time bounded video applications with a limited numbers of retransmission, APs and cards would however use traditional algorithms. Empirical results show that traditional algorithms use high PER thresholds that are not adapted to low latency requirements and switch down when the video is already heavily corrupted. The study based on video quality shows that lower PER thresholds are preferable and in this case the system switches down when the video is not yet corrupted. Practically, throughput based algorithms would switch down at a PER greater than 0.1, whereas the presented scheme switches down at a PER of around 5×10^{-2}. Similarly, the throughput based algorithms would switch up at a PER of around 5×10^{-2}, whereas the presented scheme up-scales the operating mode at a PER of around 5×10^{-4}. Results showed that traditional algorithm thresholds are not appropriate for low latency video applications, and the use of adapted PER thresholds showed a gain a 7dB in PSNR over the throughput-based algorithms. A gain of 7dB in PSNR was obtained when appropriate PER threshold tables were chosen. The PER thresholds for up and down switching should be carefully chosen for reliable video and multimedia transmission. The influence of various parameters on the PER threshold were studied, including the number of MAC ARQs, video bit rates, packet length, video content and error concealment techniques. As the number of MAC retransmission increases, the PER thresholds were seen to move towards higher PERs and were closer to the throughput-based scheme.

In chapter 8, real measurements were discussed. These measurements were taken using client/server software and IEEE 802.11a and IEEE 802.11g/b cards. Logged data was processed and cross layer results such as PER, RSSI, delay, video bit rate,

link speed extracted in order to assess the transmission performance. A comparison between UDP and TCP was performed. Thanks to its ARQ mechanism, TCP provides an error-free transmission, whereas UDP, without ARQ, can not recover missing packets above the MAC. This is however at the expense of large delays (up to 10s) due to the TCP ARQ building up ARQ delay for each retransmission. The need for flow control at the transmitter in order to limit lost packets due to bad channel conditions was highlighted. The influence of packet size has been studied in terms of PER. The packetisation studies of chapter 6 have also been investigated. The modifications of the MAC could not be implemented but the importance of a recovery scheme has been outlined when several NAL units are encapsulated into one packet for transmission. The importance of the way NAL units are mapped onto the underlying network has been stressed. The Link Adaption algorithm studied in chapter 7 has also been discussed. The proposed algorithm based on video quality could not be implemented. Instead of using PER thresholds, C/N thresholds have been used to illustrate the need of an appropriate algorithm by comparing the throughput based algorithm with the proposed video quality based scheme. 5dB of gain was obtained, stressing therefore the importance of the link adaptation strategy on the system performance and video quality.

9.2 Achievements

In this thesis, the transmission of H.264 video over IEEE 802.11a/g PHY layer was studied. A cross layer approach was adopted, where various layers interacted, coordinated their efforts, and shared information in order to provide the best video QoS. The main achievements can be summarised as follows:

- The performance of the IEEE 802.11a/g PHY layer were investigated. The low throughput efficiency of IEEE 802.11 MAC and its dependency on the packet-length dependency were presented. Small packets lead to large overheads. Moreover, the medium access scheme (CSMA/CA) and the mandatory retransmission algorithm were seen to generate delays at the application. The trade-off between PER, delay and throughput was stressed. Robust video encoding techniques were highlighted, with the focus given to the error resilient encoding features of H.264.

- In order to minimise the use of ARQ for video transmission, enhancements to the IEEE 802.11a/g PHY layer were presented, combining Space Time Block Codes and concatenated Reed Solomon Codes. Results showed a 10dB improvement for a PER of 10^{-2} when a 2Tx-2Rx STBC is combined with a RS(188,252,t=32) code correcting up to 32 bytes. Moreover, results showed that the number of ARQs is considerably reduced, improving the delay performance of the system.

- To overcome the low throughput efficiency of the IEEE 802.11 MAC without losing H.264 video quality, modifications to the legacy MAC, using packet aggregation with an error-free packet recovery mechanism, were proposed. This

allows:

- to maintain the throughput by using larger MAC frames,
- robust video transmission by using small NAL units and by recovering error-free NAL units in a corrupted MAC frame.

Results showed a gain of 3dB in PSNR for a PER of 10^{-2} with the proposed modification. In addition, the use of interleaved cross-packet FEC with an RS erasure correction code at the application layer was investigated and a gain of 6dB in PSNR was obtained with a coding rate of 0.875 and a depth of 8 packets for a PER of 10^{-2}.

- Traditional link adaptation algorithms use retransmission, optimise the error-free data throughput and do not take into account the nature of the transmitted data. In reality, they are applied even for transmission of time-bounded applications with limited of retransmissions. For low latency video applications, extensive use of retransmission is not possible and the overall received video quality has to be optimised rather than the throughput. By transmitting different video qualities on each PHY mode, empirical results showed that PER thresholds used by traditional algorithms are not appropriate for real-time video transmission, and switchings occur are high PER with the traditional schemes, i.e. when the video is already heavily corrupted. Lower PER thresholds are preferable. 7dB gain in PSNR over throughput-based algorithms were noticed when using adapted thresholds. Influences of video content, number of retransmissions, packet length, and video bit rate were discussed.

- Real measurement data was collected for IEEE 802.11a/b/g at the PHY, MAC and transport layers. TCP and UDP links were studied. The lack of flow control was highlighted as a weakness of the UDP protocol. Data was used to simulate video transmission. The influence of packet length on the system performance was investigated. A study was conducted in order to stress the need for a smart packetisation strategy. Moreover, PSNR results showed a 5dB in PSNR gained when an appropriate link adaptation algorithm is employed. It clearly shows that, with traditional algorithms, the video applications with limited numbers of ARQs break up whenever the mode is about to change. The suggested algorithm smooths transitions, and thus enables mobility.

9.3 Suggestions for Future Work

There are numerous extensions to the work presented in this thesis.

- Following the discussion in chapter 7, a rigorous algorithm could be developed in order to define the PER thresholds for a video quality-based link adaptation scheme in the context of video transmission requiring a limited number of MAC ARQ.

- IEEE 802.11g and b operate in the ISM 2.4GHz band. Due to its global availability, popularity and suitability for low cost radios [183], the 2.4GHz band is now overcrowded. In addition to IEEE802.11g/b, the Wireless Personal Area Networks (WPAN) IEEE 802.15.1 [184] and Bluetooth [185, 186, 187], cordless telephones such as DECT as well as microwave ovens are also operating in this band. If deployed in the same environment, interferences may lead to significant performance degradation. Further work could concentrate on the impact of uncoordinated interference on video transmission in these ISM bands.

- In the near future, IEEE 802.11n will complete its specification process. One of the main features of IEEE 802.11n is that it should implement a PHY layer supporting up to 100Mbits/s. Multiple-Input-Multiple-Output (MIMO) will be used to reach such high rates. To support the PHY layer, the MAC shall be carefully designed. Because of the very poor overhead efficiency of the IEEE 802.11 MAC, delivery at such speeds can not be guaranteed and a more efficient MAC, derived from 802.11e, is required.

- Following the discussion of chapter 8, further processing of data could be investigated. Error models based on real-data measurements could be developed for various scenarios. Moreover, cross layering interactions could be investigated with the video coding adapting to channel bandwidth variations due to congestion. In order to support bandwidth reduction, the video encoder could reduce its bit rate by adapting the quantiser so that the received video quality does not degrade dramatically. Transcoding is another possible solution to avoid network congestion.

- As mentionned in chapter 8, the UDP transport layer does not allow flow control from the wired/ethernet world to the wireless world. In order to avoid data overflow when the channel conditions turn bad, a MAC level buffer could be developed to operate under the UDP protocol stack.

- Further work can combine scalable video encoding and WLANs. H.264 is currently developing scalability features to be incorporated in the standard. Data partitioning is already supported. IEEE 802.11a/g provides different operating modes with different reliabilities. Base layer data containing the most important information shall therefore be transmitted on the most robust modes in order to guarantee a basic video quality. Enhancement layers containing refinement information do not need such protection and can be transmitted with less resilient modes.

- A similar study could be performed with the use of the Metropolitan Area Network (MAN) IEEE 802.16, also known as Worldwide Interoperability for Microwave Access (WiMAX), providing data rate up to 70Mbits/s in the 10-60GHz range.

Appendix A

IEEE 802.11a/g and Hiperlan/2 PHY layer Description

A.1 Convolutional Encoder with 1/2 rate

The scrambled data are input to the convolutional encoder. It translates blocks of k bits into blocks of n bits. The output of the convolutional encoder depends on the input block of k binary elements and also on the m blocks present in the memory (shift register), as shown in figure A.1 [140, 188]

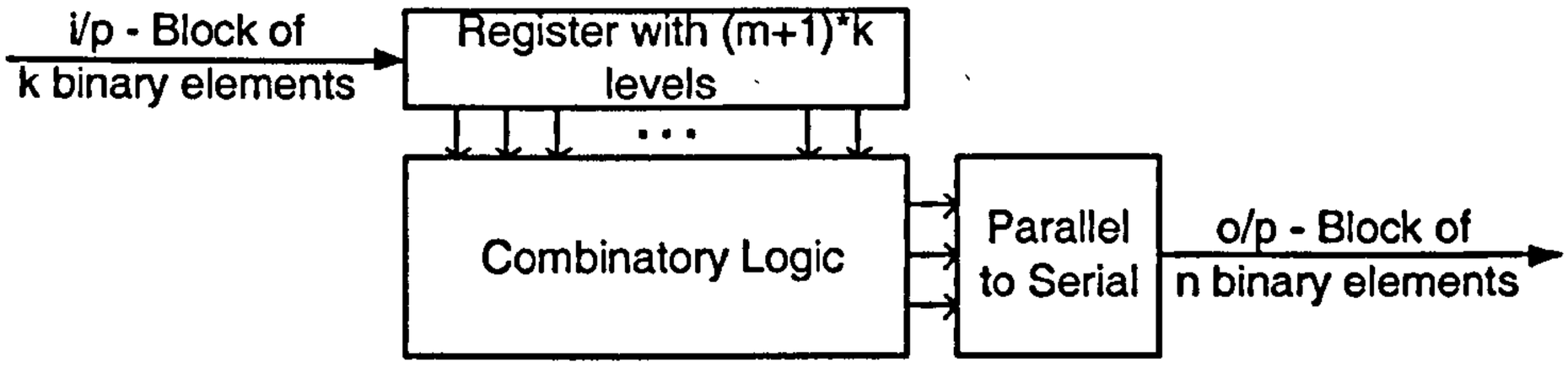


Figure A.1: Principle of a convolutional encoder

The register memorises the *(m+1)* blocks of k elements, the first block in the shift register being the current input. *(m+1)* is the constraint length of the encoder. The rate of the decoder is defined by: $R = \frac{k}{n}$. In the case of Hiperlan/2 and IEEE802.11a, the encoder is 1/2 rate with a constraint length of 7 (m=6). Thus $n = 2 \times k$ bits are output for every k input bits. Figure A.2 shows the mother convolutional code with a rate of 1/2 used in Hiperlan/2 and IEEE802.11a.

T_b represents a bit delay. The output X and Y are generally serialised after puncturing. The sequence generator for X and Y are respectively: $(133)_8 = (1011011)_2$ and $(171)_8 = (1111001)_2$ [3, 4, 6]. Thus the generator polynomials for X and Y are respectively:

$$G_X(D) = 1 + D^2 + D^3 + D^5 + D^6 \quad \text{(A.1)}$$

$$G_Y(D) = 1 + D + D^2 + D^3 + D^6 \quad \text{(A.2)}$$

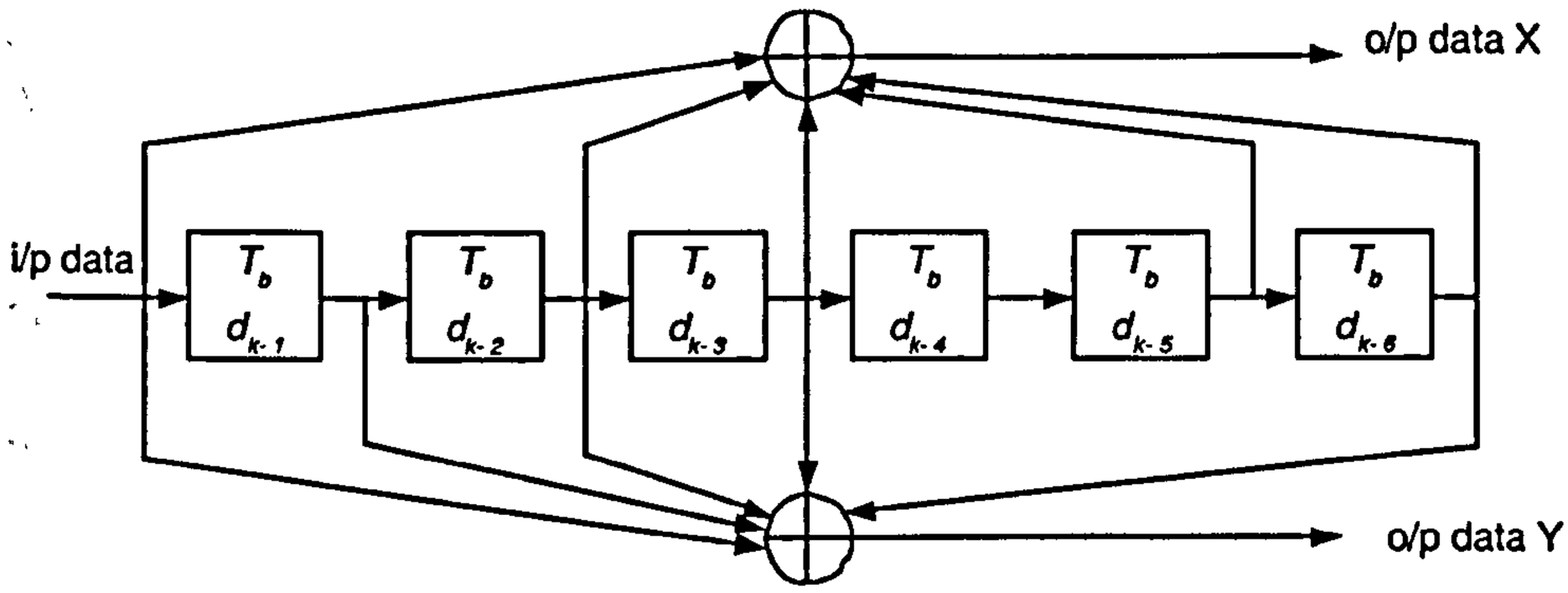


Figure A.2: Mother Convolutional Code (1/2 rate)

If $\{d_k\}$ represent the values in the register, d_k are the current input, X and Y can be expressed as:

$$X = d_k + d_{k-2} + d_{k-3} + d_{k-5} + d_{k-6} \tag{A.3}$$

$$Y = d_k + d_{k-1} + d_{k-2} + d_{k-3} + d_{k-6} \tag{A.4}$$

The decoding process at the receiver uses the Viterbi Algorithm and is based on minimum distance: the received codeword is compared with each of the possible codewords in a treillis diagram and the closest to the received codeword is chosen. In order to restore the shift register to its all zero intial state for the next message, m zeros bits are appended to the tail of the current message. For Hiperlan/2 and IEEE802.11a, 6 bits are appended to the message before convolutional encoding.

A.2 Puncturing

The puncturing is performed in two stages: *P1* and *P2*. Given the use of a 1/2 rate code, the 6 tail bits generate 12 bits. Because Hiperlan/2 uses a fixed size of 54 bytes, these 12 bits are removed with *P1*. IEEE 802.11a/g does not perform *P1* since it allows variable length packets. *P1* is shown in figure A.3.

X	X_1	X_2	X_3	X_4	X_5	X_6		X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}
Y	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	

Figure A.3: *P1* puncturing pattern

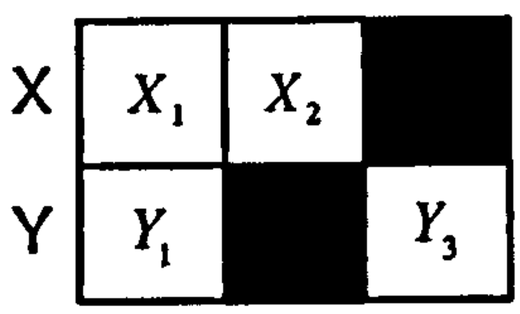
This puncturing uses the parallel output data (X and Y) of the convolutional encoder and discards bits X_7 and Y_{13} from 26 incoming bits. This puncturing pattern removes two bits and is therefore performed 6 times so that 12 bits are discarded. After

conversion to a serial output, the 26 bits are organised as:
$X_1Y_1X_2Y_2X_3Y_3X_4Y_4X_5Y_5X_6Y_6Y_7X_8Y_8X_9Y_9X_{10}Y_{10}X_{11}Y_{11}X_{12}Y_{12}X_{13}$

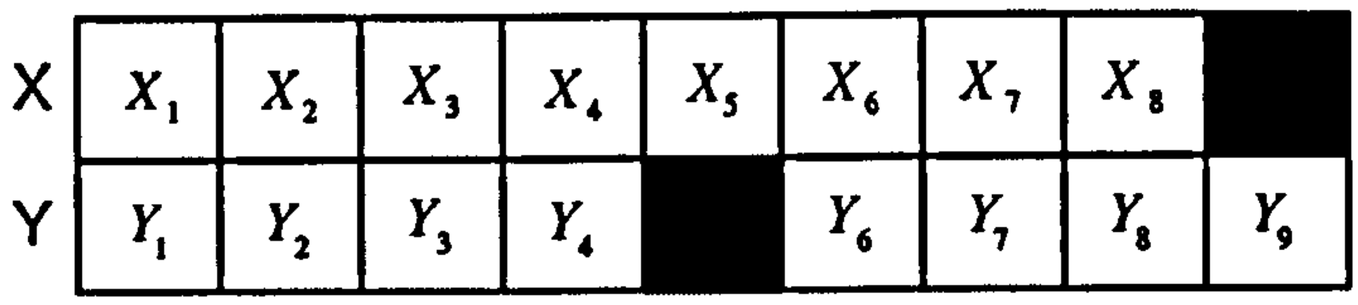
In order to achieve the 3/4, 9/16 and 2/3 rate codes, the second puncturing stage *P2* is required. Data are first converted to a parallel form using the output of the serialised first puncturing stage. Then, different puncturing patterns are applied depending on the coding rate [3, 4, 6] as shown in figure A.4. The corresponding outputs for these puncture schemes are (after serialisation):

- For coding rate 3/4: $X_1Y_1X_2Y_3$
- For coding rate 9/16: $X_1Y_1X_2Y_2X_3Y_3X_4Y_4X_5X_6Y_6X_7Y_7X_8Y_8Y_9$
- For coding rate 2/3: $X_1Y_1X_2$

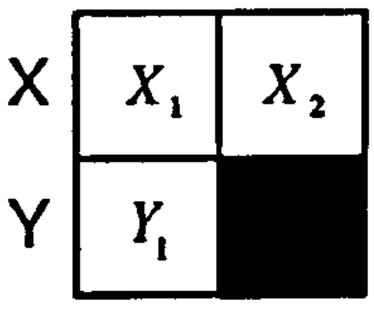
Figure A.5 summarises the whole channel encoding process, including the puncturing.



(a) Coding Rate 3/4



(b) Coding Rate 9/16



(c) Coding Rate 2/3 (IEEE only)

Figure A.4: Puncturing Patterns *P2*

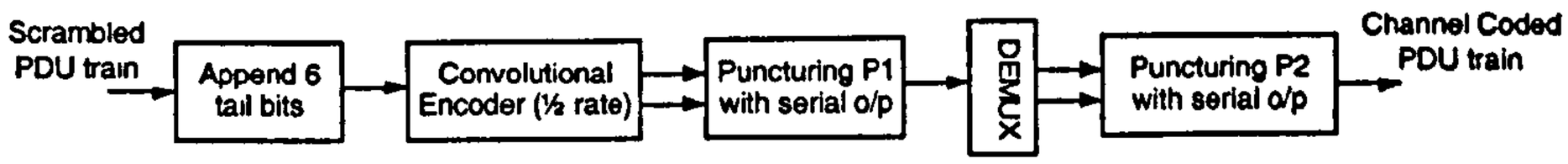


Figure A.5: Channel Coding and Puncturing process

A.3 Interleaving

To prevent burst errors from being input to the decoder at the receiver, data are interleaved. The decoding process uses the Viterbi algorithm with soft decisions, which works better with random errors. Coded data are interleaved with a block interleaver [3, 4, 6]. Coded data enter row by row into a memory buffer and when the buffer is filled, data are read column by column.

If an error burst occurs and corrupts successive symbols sent over the channel, the de-interleaver spreads the errors among several coded symbols. The size of the buffer depends on the size (in bits) of an OFDM symbol (see section 2.2.2). The interleaving process in both standards is defined by a two-step permutation. The first permutation ensures that adjacent bits are mapped onto non-adjacent sub-carriers. The second permutation ensures that adjacent bits are mapped alternatively onto less and more significant bits, since bits in QAM do not have the same error probabilities [26, 44].

A.4 Mapping

Interleaved data enter the 'mapper' and are divided into groups of N_{BPSC} bits, where N_{BPSC} represents the number of Bits Per Sub-Carriers [3, 4, 6]. Depending on the modulation used, N_{BPSC} takes different values:

- $N_{BPSC} = 1$ bit for a BPSK modulation
- $N_{BPSC} = 2$ bit for a QPSK modulation
- $N_{BPSC} = 4$ bit for a 16QAM modulation
- $N_{BPSC} = 6$ bit for a 64QAM modulation

Each group is then converted into a complex number $I+jQ$ representing the symbol from the corresponding modulation constellation. The conversion is performed according to a *Gray* coded constellation map. Table A.1 [3, 4, 6] illustrates the different mapping from input bits to the I and Q values for all modulation schemes. A normalisation factor K_{MOD} is used to achieve the same average power for all mappings. Table A.2 gives the different values of K_{MOD}. The output values d have the following form:

$$d = (I + jQ) \times K_{MOD} \tag{A.5}$$

The 'mapper' generates symbols of the general form: $A_n e^{j\phi_n}$

Table A.1: Encoding Tables for Mapping

(a) BPSK

Input bit b_1	*I-out*	*Q-out*
0	-1	0
1	1	0

(b) QPSK

Input bit b_1	*I-out*	*Input bit* b_2	*Q-out*
0	-1	0	-1
1	1	0	1

(c) 16 QAM

Input bit b_1b_2	*I-out*	*Input bit* b_3b_4	*Q-out*
00	-3	00	-3
01	-1	01	-1
11	1	11	1
10	3	10	3

(d) 64 QAM

Input bit $b_1b_2b_3$	*I-out*	*Input bit* $b_4b_5b_6$	*Q-out*
000	-7	000	-7
001	-5	001	-5
011	-3	011	-3
010	-1	010	-1
110	1	110	1
111	3	111	3
101	5	101	5
100	7	100	7

Table A.2: Modulation dependent normalisation factor

Modulation	K_{MOD}
BPSK	1
QPSK	$1/\sqrt{2}$
16 QAM	$1/\sqrt{10}$
64 QAM	$1/\sqrt{42}$

Appendix B

Hiperlan/2 PHY Burst and Medium Access Control

Hiperlan/2 implements three types of preambles for the physical burst (*A*, *B* and *C*) depending on the type of burst) and five types of bursts (Broadcast burst, Downlink burst, Uplink burst with short preamble, Uplink burst with long preamble, Direct link burst) [6]. Hiperlan/2 transmits PDU bursts or PDU trains coming from the MAC Layer. A PDU train is composed of several concatenated PDUs. Two types of PDUs are defined: short PDUs (C PDUs of 9 bytes) and long PDUs (U PDUs of 54 bytes). Figure B.1 shows the long PDU format. The 54 bytes are composed of 49.5 bytes for the Data Link Control (DLC) Data Unit, 1.5 bytes for header and 3 bytes for a Cyclic Redundancy Check (CRC) used for error detection. The 49.5 bytes of the DLC data unit include 48 bytes payload and 1.5 bytes of Flags [44].

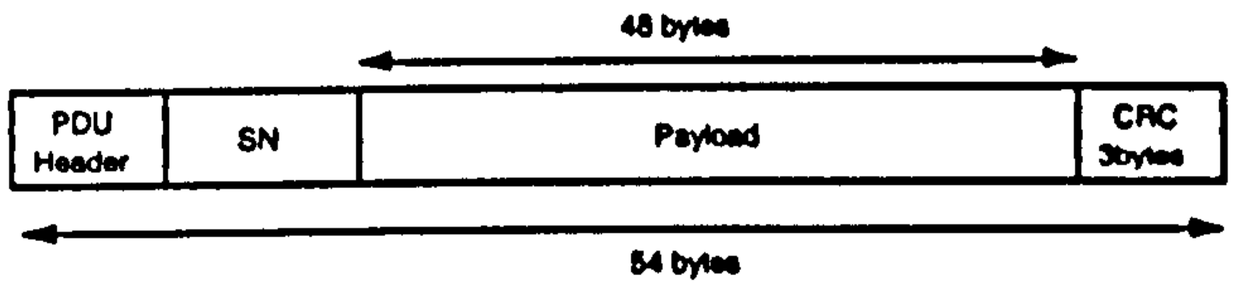


Figure B.1: Format of a long PDU in Hiperlan/2

The protocol used in Hiperlan/2 is TDD/TDMA (Time Division Duplex/Time Division Multiple Access) and works in a synchronous mode [6, 67]. It provides therefore a guaranteed bandwidth with a contention free channel access. The MAC frame has a time-slotted structure. Communications are simultaneous in both downlink and uplink within the same frame. A MAC frame has a fixed duration of 2ms as shown in figure B.2 and is composed of: Broadcast CHannel (BCH), Frame CHannel (FCH), Access Feedback CHannel (ACH), Random Access CHannel (RCH) and Down Link phase (DL), Uplink phase (UL), Direct Link phase (DiL). DLs, ULs and DiLs consist of PDU trains and each user is assigned a slot in the DiL, UL or DL phase where it is allowed to transmit.

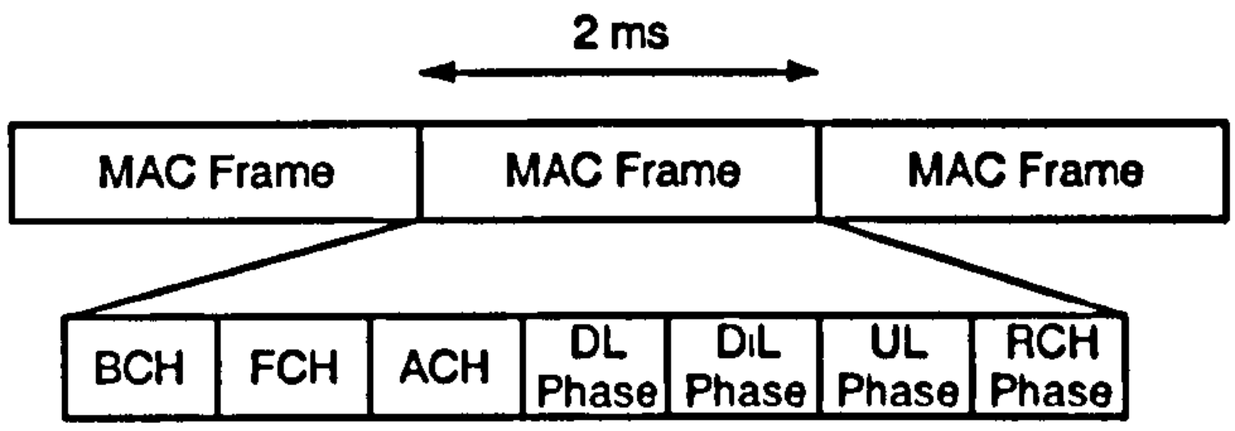


Figure B.2: Hiperlan/2 MAC frame

B.1 ARQ and Delay study

Hiperlan /2 implements a selective repeat ARQs and ARQ delays are defined in terms of MAC frames. The extra ARQ processing delay in the transmit direction (d_{Tx}) is related to the minimum number of MAC frames the transmitter must wait to retransmit the packet once it receives at ARQ request. The extra ARQ processing delay in the receive direction (d_{Rx}) is the minimum number of MAC frames the receiver has to wait before it produces an ARQ feedback message. The round trip time (r_{tt}) in the downlink (DL) and uplink (UL) direction (in terms of the number of MAC frames) is given by [6]:

$$r_{tt} = d_{Tx} + d_{Rx} + 1 \tag{B.1}$$

The round trip delay denotes the delay the receiver must wait for a packet in error to be resent using one ARQ cycle. Hiperlan /2 defines 4 classes of ARQ delay and these are shown in table B.1 [6].

Table B.1: Hiperlan/2 ARQ delay Class

Delay Class	d_{Tx}, d_{Rx} *(in MAC frames) for* r_{tt} *calculation*	*Required Maximum Processing delay*
0	0	40μs
1	1	2ms
2	2	4ms
3	3	6ms

Figure B.3 gives an example of the ARQ scheme with a class 1 delay. For this case, r_{tt} is 3 frames long. The delay the receiver must wait after one ARQ to get a packet retransmission is $r_{tt} + 1$ MAC frames, i.e. 8ms. For two and three ARQs, the delay increases to 16ms and 24ms respectively. In general terms, assuming n ARQs are required then the delay will be 8*n ms. Similarly, for a delay of class 2, the r_{tt} is equal to 5 MAC frames and the receiver must wait 12ms for the first ARQ, 24ms after two ARQs and 36 ms after three ARQs. In general terms, assuming n ARQs the delay will be 12*n ms. These delays are shown in table B.2.

Figure B.4 compares the delay for up to 10 ARQ cycles for the 4 ARQ classes defined in Hiperlan /2. There should also be a very small offset due to the channel

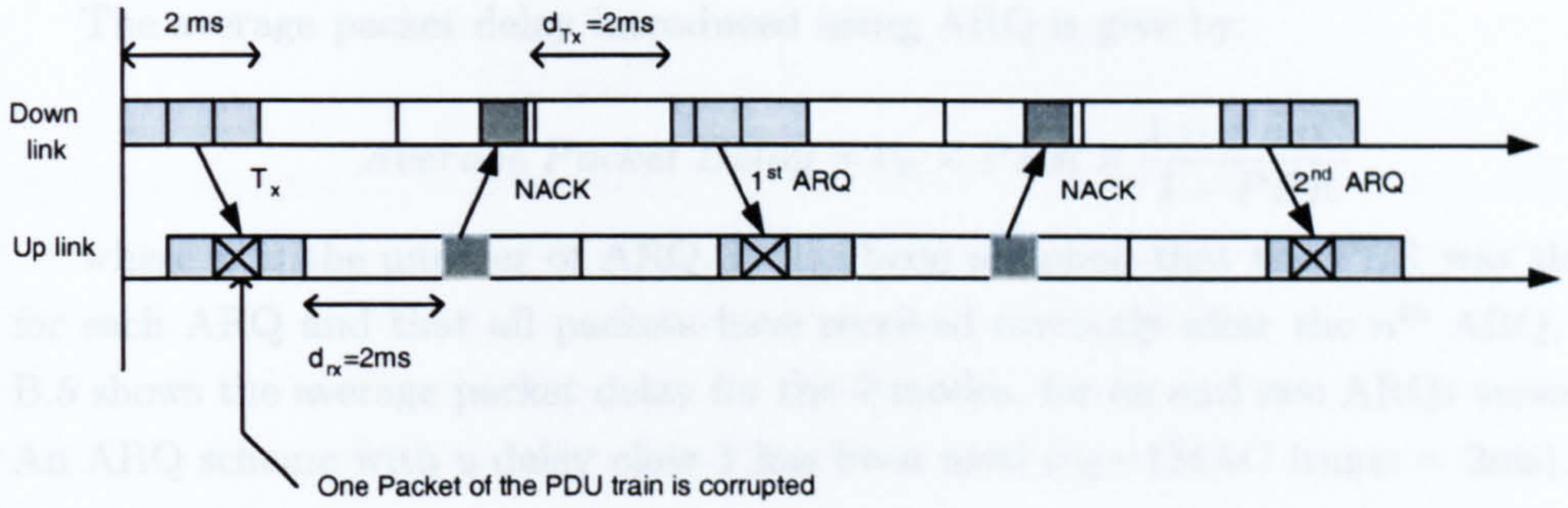


Figure B.3: ARQ in Hiperlan/2 with delay of Class 1

Table B.2: ARQ Packet delays

Delay Class	r_{tt} *in MAC frames)*	*Delay after n ARQ:* δ *in ms*
0	1	2n
1	3	8n
2	5	12n
3	7	16n

propagation delay. However, as mentioned earlier this is ignored since it lies in the order of tens of nanoseconds. We see that Class 0 devices show the best ARQ delay. Class 0 devices limit the time to process ARQ signals to 40 s. The other three classes' limit this time to 2ms, 4ms and 6ms respectively.

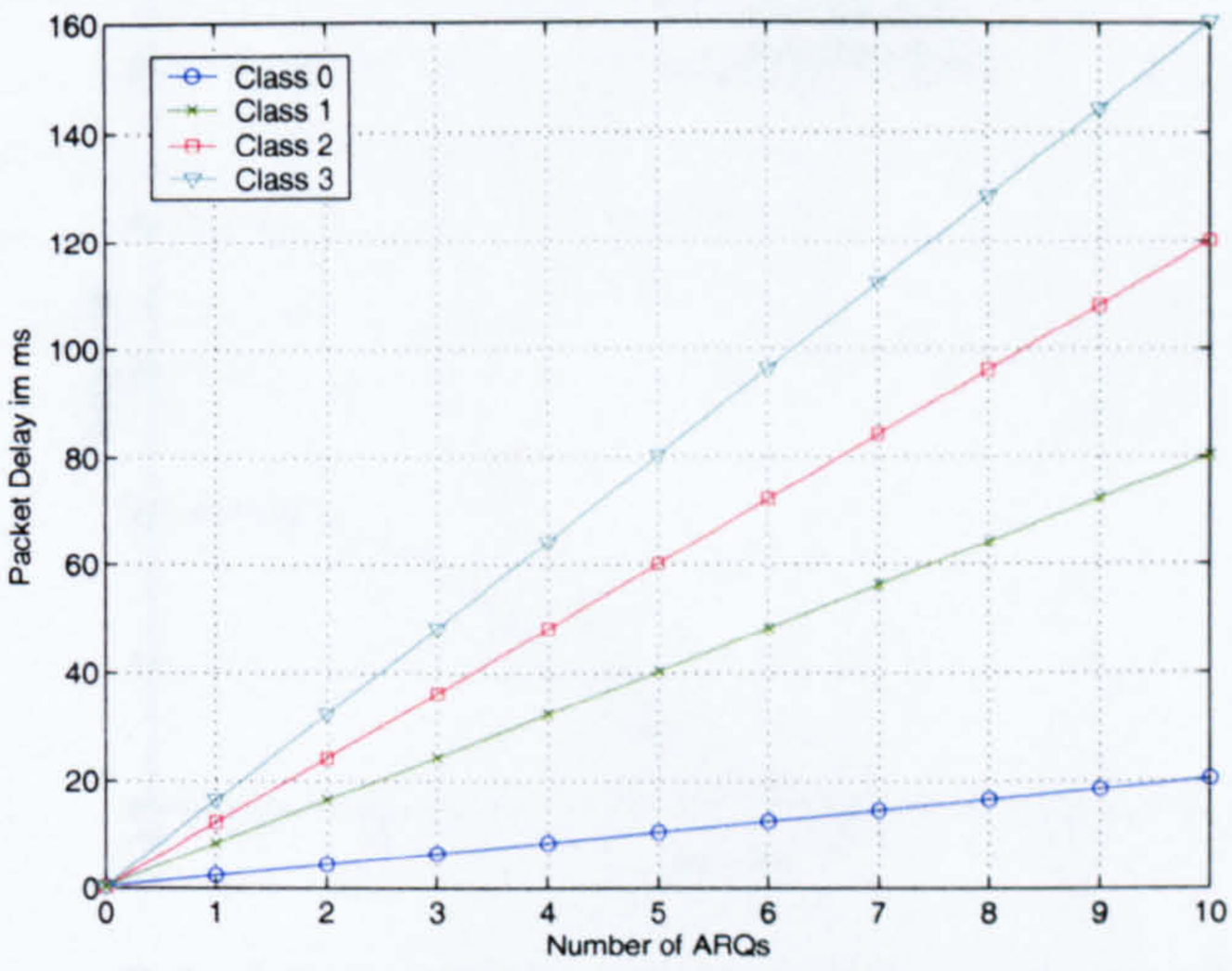


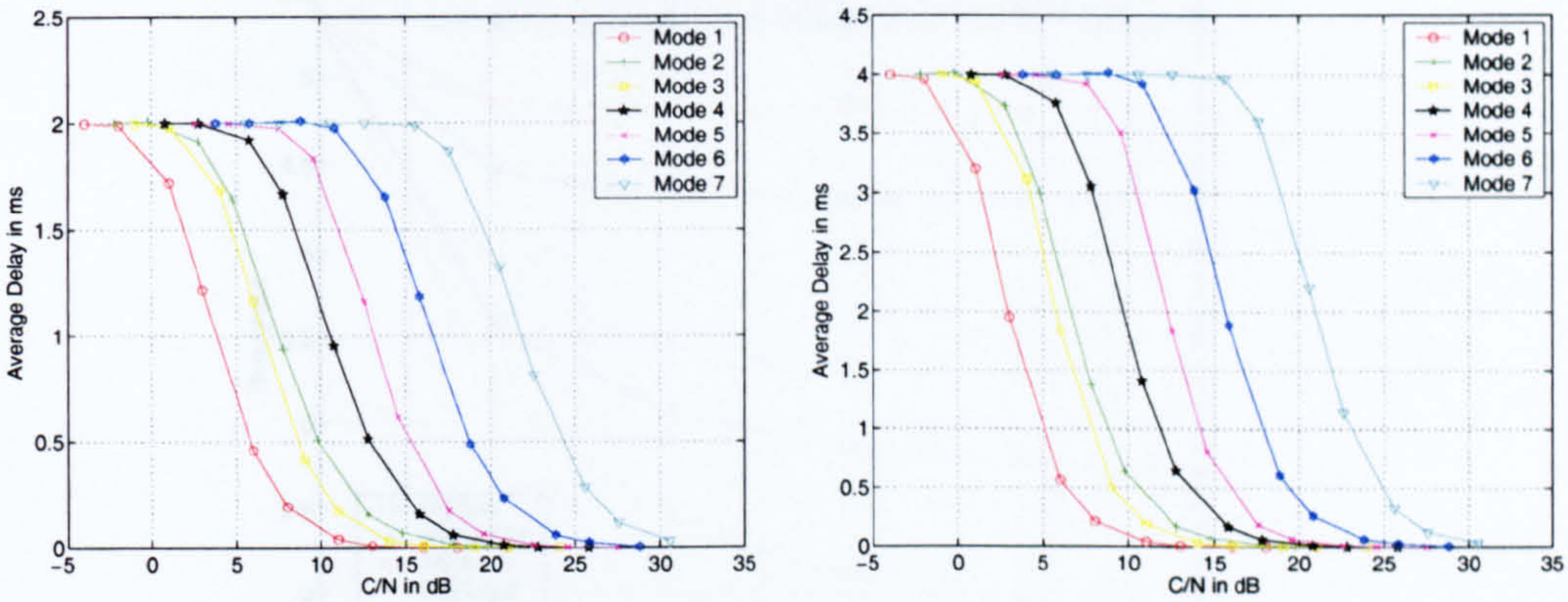
Figure B.4: ARQ classes-delay comparison

If, for example, an ARQ latency of 50ms is required, it can be seen than a class 0 device achieves a maximum of 25 retransmissions. Class 1, 2 and 3 devices can achieve 8, 4 and 2 retransmission in this time window. Clearly it is important for a low delay class to be specified for a real-time video transmission.

The average packet delay introduced using ARQ is give by:

$$Average\ Packet\ Delay = r_{tt} \times PER \times \frac{1 - PER^n}{1 - PER} \tag{B.2}$$

where n is the number of ARQ. It has been assumed that the PER was the same for each ARQ and that all packets have received correctly after the n^{th} ARQ. Figure B.5 shows the average packet delay for the 7 modes, for on and two ARQs versus C/N. An ARQ scheme with a delay class 1 has been used (r_{tt}=1MAC frame = 2ms).



(a) One ARQ (b) Two ARQs

Figure B.5: Average Packet Delay Mode Comparison

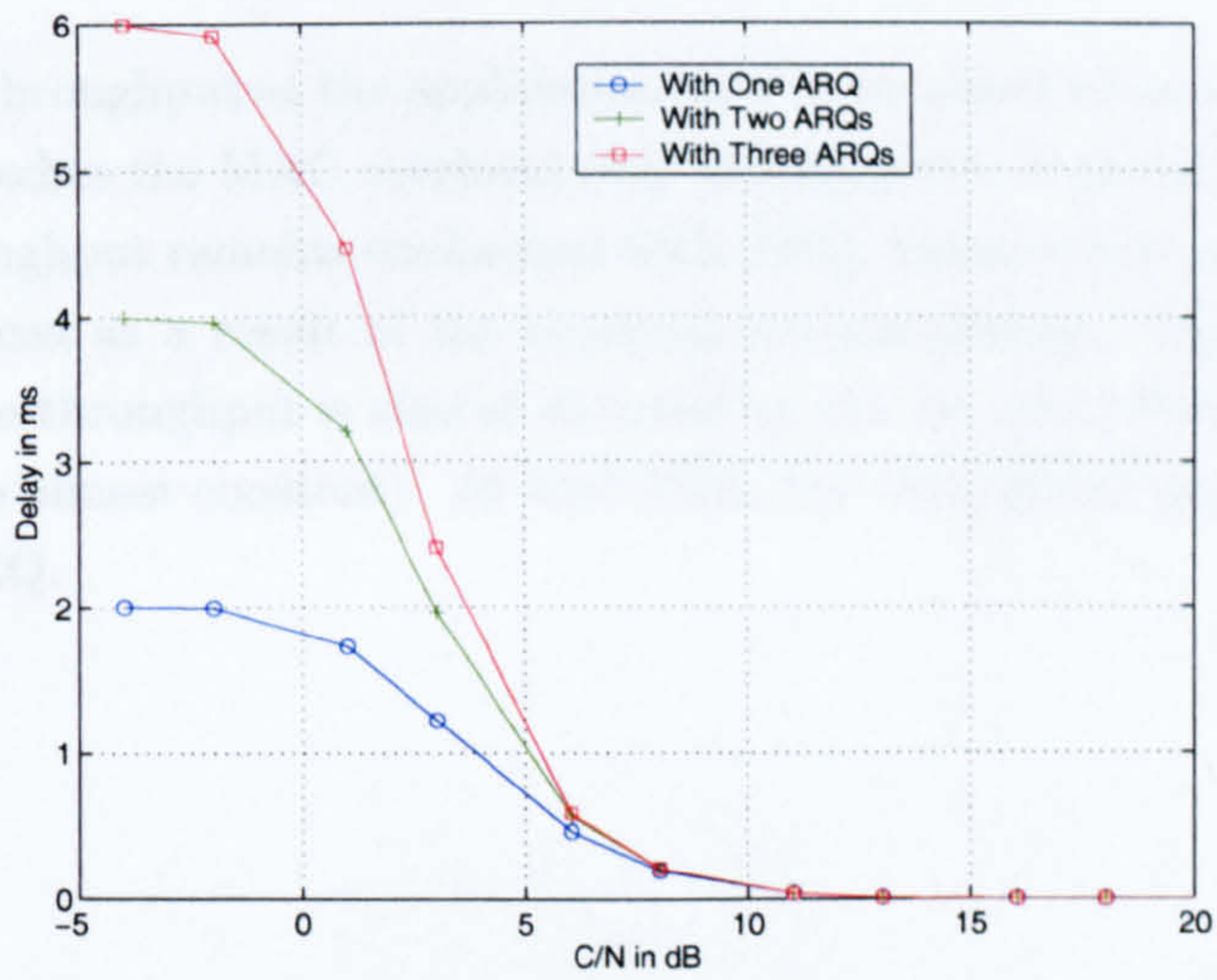


Figure B.6: Average Packet Delay ARQ comparison for mode 1

B.2 ARQ and Throughput study

The throughput at the video application is also a function of the ARQ strategy and is given by (for n ARQs):

$$Throughput = R \times \frac{1 - PER}{1 - PER^{n+1}} \tag{B.3}$$

where R represents the nominal throughput, taking the MAC overheads into account (48 byte payload in 54 bytes packets). The throughput performance is expected to be higher with fewer ARQs. The greater the number of ARQs, the more of the bandwidth that is used for retransmissions and the longer the latency. Figure B.7 shows the throughput, with different ARQs, for Hiperlan /2 mode 1.

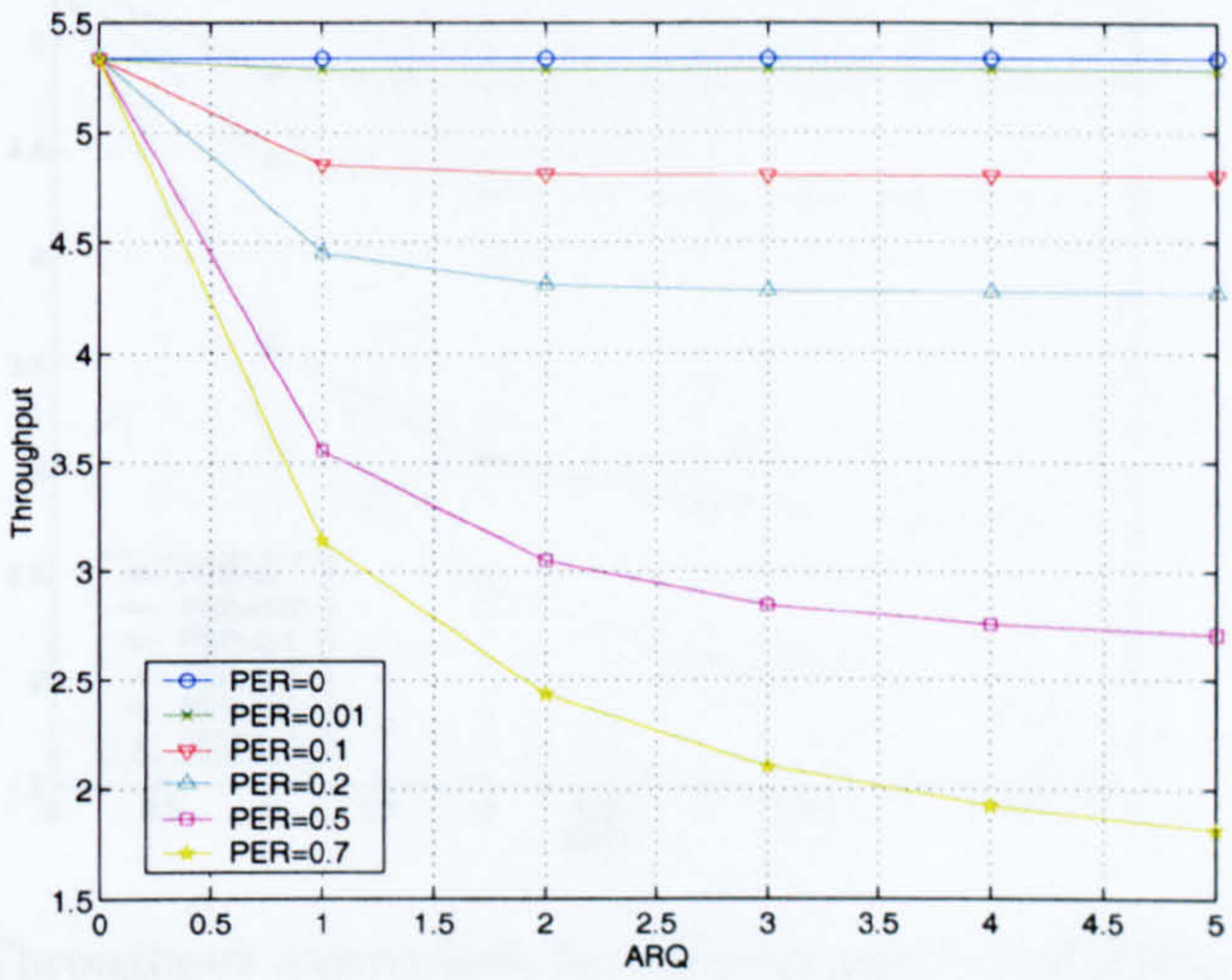


Figure B.7: Throughput comparison for different number of ARQs, Mode 1

The maximum throughput at the application layer is obtained when no ARQ is used (where the value reaches the MAC overhead only throughput). It should be noted that the PHY layer throughput remains unchanged with ARQ, however the application layer throughput is reduced as a result of the required retransmissions. Figure B.7 shows that at low PER the throughput is almost dictated by the 1st ARQ (beyond which the throughput remains almost constant). At high PER, the throughput degrades strongly as a function of ARQ.

Appendix C

Markov Chain Study of the IEEE802.11 MAC

This appendix details the study of Markov chains applied to IEEE 802.11 MAC with several users, and is mainly based on [25, 87, 88, 89]. This study has been used in section 3.4.

We consider n contending stations and we assume the saturation condition, i.e. the transmission queue is always non-empty. We also assume that there is no hidden station. We define:

- σ: the slot time.
- $T_{success}$: Average time the channel is sensed busy due to one station transmitting successfully
- $T_{collision}$: Average time the channel is sensed busy due one collision occurring.
- $b(t)$: the back-off time counter for a given station at time t.
- $s(t)$: the 'back-off stage' of a given station a time t ($s(t)$=i meaning that the CW size is $W_i = 2^i \times CW_{min}$).
- $\{s(t), b(t)\}$: a bi-dimensional process with discrete-time Markov chain as depicted in figure C.1.
- τ: the probability that a station transmits a packet in a generic slot time.
- p: the probability of collision seen by a packet being transmitted on the channel.
- $P\{i_1, k_1/i_0, k_0\} = P\{s(t+1) = i_1, b(t+1) = k_1/s(t) = i_0, b(t) = k_0\}$ is the probability of having:
  - CW size of $2^{i_1} \times CW_{min}$ at time *t+1*
  - Back-off Counter of k_1 at time *t+1*

  knowing that:
  - CW size of $2^{i_0} \times CW_{min}$ at time t

– Back-off Counter of k_0 at time t

C.1 Probabilities of Transmission and Collision

With these definitions, $P\{i,k/i,k+1\}$ is the probability of having: $CW = 2^i \times CW_{min}$ and *counter* $= k$ at time *t+1* knowing that $CW = 2^i \times CW_{min}$ and *counter* $= k+1$ at time *t*. At the beginning of each slot, the back-off counter is decremented from k+1 (at t+1) to k (at t). This leads to:

$$\boxed{P\{i,k/i,k+1\} = 1 \text{ for } k \in (0, W_i - 2) \text{ and } i \in (0,m)} \quad \text{(C.1)}$$

$P\{0,k/i,0\}$ is the probability of having $CW = CW_{min} = W_0$ and *counter* $= k$ at time *t+1* knowing that $CW = 2^i \times CW_{min}$ and *coutner* $= 0$ at time *t*. A new packet following a successful packet transmission starts with a back-off stage of CW_{min} and the counter is uniformely chosen in the range $[0, W_0 - 1]$. The probability of a packet transmitted without collision is *1-p*. The probability of choosing k in $[0, W_0 - 1]$ is $\frac{1}{W_0}$. Therefore:

$$\boxed{P\{0,k/i,0\} = \frac{1-p}{W_0} \text{ for } k \in (0, W_i - 0) \text{ and } i \in (0,m)} \quad \text{(C.2)}$$

$P\{i,k/i-1,0\}$ is the probability of having $CW = 2^i \times CW_{min}$ and *counter* $= k$ at time *t+1* knowing that $CW = 2^{i-1} \times CW_{min}$ and *counter* $= 0$ at time *t*. Because at *t*, the counter is equal to 0 and that the CW is incremented, we know that a collision has occurred. The probability of a collision occurring is p, and the probability of choosing k in $(0, W_{i-1})$ is W_i. Therefore:

$$\boxed{P\{i,k/i-1,0\} = \frac{p}{W_i} \text{ for } k \in (0, W_i - 1) \text{ and } i \in (1,m)} \quad \text{(C.3)}$$

$P\{m,k/m,0\}$ is the probability of having $CW = 2^m \times CW_{min} = CW_{max}$ and *counter* $= k$ at time *t+1* knowing that $CW = 2^m \times CW_{min}$ and *counter* $= 0$ at time *t*. Similarly, we deduce that a collision has occurred at the stage m. The back-off stage can not be increased, so we stay in stage m. Therefore:

$$\boxed{P\{m,k/m,0\} = \frac{p}{W_m} \text{ for } k \in (0, W_m - 1)} \quad \text{(C.4)}$$

Let $b_{i,k} = lim_{t\to\infty} P\{s(t) = i, b(t) = k\}, i \in (o,m), k \in (0, W_i - 1)$ be the stationary distribution of the chain. We can work out that the following relationship:

$$b_{i,0} = p \times b_{i-1,0} = p \times b_{i-1,0} = p^i \times b_{0,0} \; for \; i \in [0, m-1[\quad \text{(C.5)}$$

$$b_{m,0} = \frac{p^m}{1-p} \times b_{0,0} \quad \text{(C.6)}$$

The regularities of the chain allows us to derive:

$$b_{i,k} = \frac{W_i - k}{W_i} \times p \times b_{i-1,0} \; for \; i \in]0, m[\tag{C.7}$$

$$b_{o,k} = \frac{W_0 - k}{W_0} \times b_{0,0} \tag{C.8}$$

$$b_{m,k} = \frac{W_m - k}{W_m} \times b_{m,0} \tag{C.9}$$

Thanks to C.5 and C.6, C.6, C.7 and C.8 can be rewritten as:

$$b_{i,k} = \frac{W_i - k}{W_i} \; for \; i \in [0, m] \; and \; k \in [0, W_i - 1] \tag{C.10}$$

The normalisation property gives us:

$$\sum_{i=0}^{m} \sum_{k=0}^{W_i - 1} b_{i,k} = 1 \tag{C.11}$$

which is developed in

$$b_{0,0} = \frac{2 \times (1 - 2p) \times (1 - p)}{(1 - 2p) \times (CW_{min} + 1) + p \times CW_{min} \times (1 - (2p)^m)} \tag{C.12}$$

For a collision to occur, at least one of the (n-1) other stations need to transmit in the same time slot. This gives us:

$$\boxed{p = 1 - (1 - \tau)^{n-1}} \tag{C.13}$$

Transmissions only occurs in stage (i, 0) for $i \in [0, m]$, .i.e. when the counter has reached zeros. If τ defines the probability that a station transmit in a generic time slot, we can work out:

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{b_{0,0}}{1 - p} \tag{C.14}$$

C.12 and C.14 are combined into

$$\boxed{\tau = \frac{2 \times (1 - 2p)}{(1 - 2p) \times (CW_{min} + 1) + p \times CW_{min} \times (1 - (2p)^m)}} \tag{C.15}$$

C.13 and C.15 define a non linear system with two equations and two unknown: p the probability of collision occurring and τ the probability that a station transmit in a random slot.

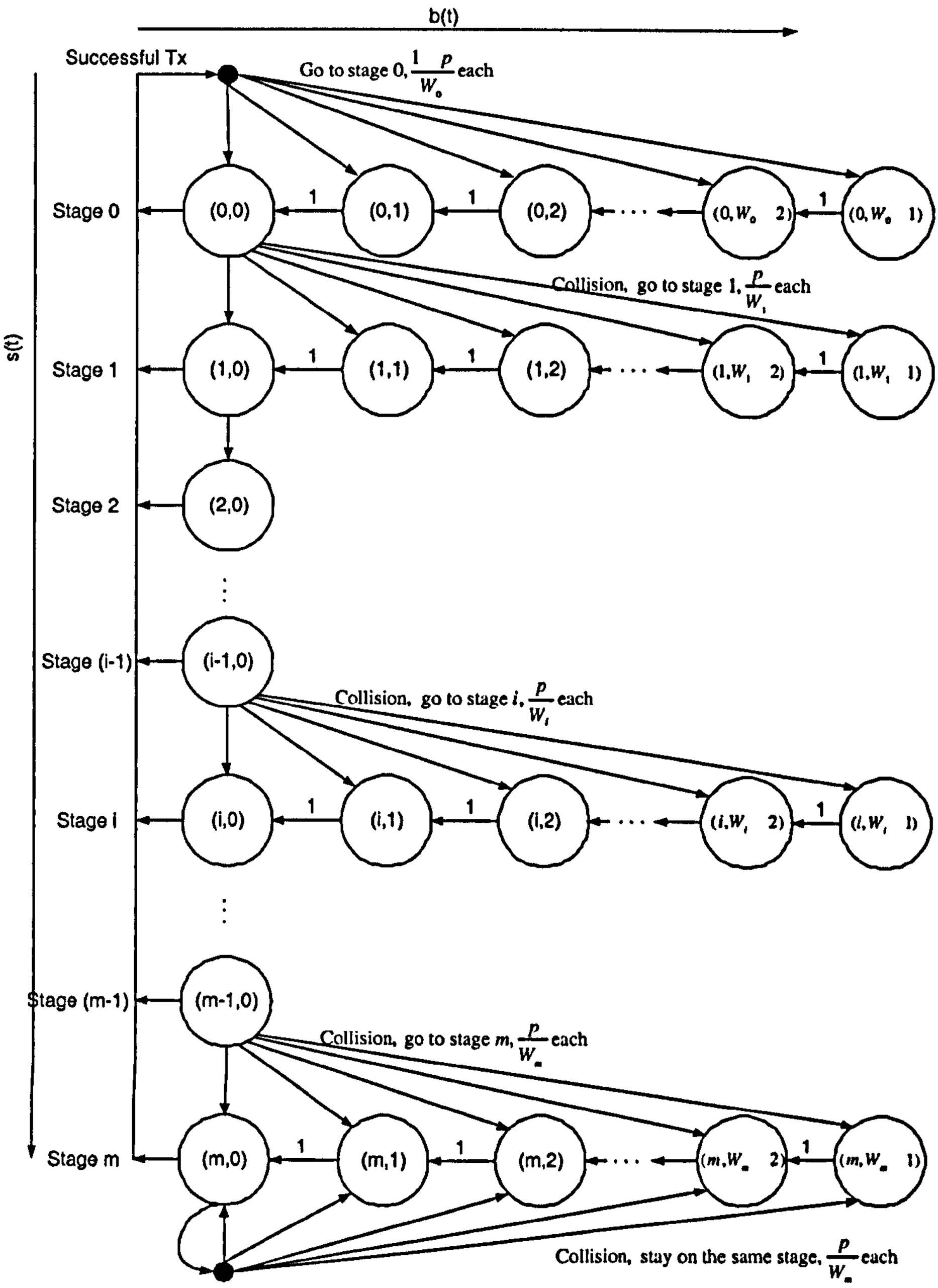


Figure C.1: Markov Chain Model for the back-off window size

C.2 Performances Derivation

C.2.1 Normalised Throughput

The probability that a station does not transmit in a generic slot time is therefore $1-\tau$, and the probability that any station transmits in a generic slot time $(1-\tau)^n$. The probability that at least one transmission occurs in the slot time is then:

$$P_{Tx} = 1-(1-\tau)^n \tag{C.16}$$

The probability that an occurring transmission is successful (P_s) is the probability that a station transmits and the remaining *n-1* remain silent (otherwise there is collision) conditioned on the fact that at least one station transmits, i.e.:

$$P_s = \frac{n \times \tau \times (1-\tau)^{n-1}}{P_{Tx}} = \frac{n \times \tau \times (1-\tau)^{n-1}}{1-(1-\tau)^n} \tag{C.17}$$

Let S be the normalised throughput defined as the fraction of time used to successfully transmit payload bits. S is expressed as:

$$S = \frac{E\left[Payload\ Information\ Transmitted\ in\ a\ Slot\ Time\right]}{E\left[Length\ of\ a\ Slot\ Time\right]} \tag{C.18}$$

$P_{Tx} \times P_s$ is the probability of a successful transmission in a slot time. If $E[P]$ denotes the average packet payload size. $E\left[Payload\ Information\ Transmitted\ in\ a\ Slot\ Time\right]$ can be expressed as:

$$E\left[Payload\ Information\ Transmitted\ in\ a\ Slot\ Time\right] = P_{Tx} \times P_s \times E[P] \tag{C.19}$$

$E\left[Length\ of\ a\ Slot\ Time\right]$ is composed of:

- Proportion of time where the slot time is empty: $(1-P_{Tx}) \times \sigma$.
- Proportion of time where the slot time is sensed busy because of a successful transmission: $P_{Tx} \times P_s \times T_{success}$
- Proportion of time where the slot time is sensed busy because of a collision: $P_{Tx} \times (1-P_s) \times T_{collision}$

C.18 can therefore be re-written as:

$$S = \frac{P_{Tx} \times P_s \times E[P]}{(1-P_{Tx}) \times \sigma + P_{Tx} \times P_s \times T_{success} + P_{Tx} \times (1-P_s) \times T_{collision}} \tag{C.20}$$

$T_{success}$ and $T_{collision}$ are mode access dependent (basic access or RTS/CTS access) and are given in equation 3.5 and 3.7 for the basic access and in equations 3.6 and 3.8 for the RTS/CTS access respectively.

C.2.2 Average Delay

The time needed for a frame to be transmitted is considered to start when a frame becomes head of the station's queue and is ended when an positive acknowledgment is received. The average frame delay can expressed as:

$$E[D] = E[x] \times E[Length\ of\ a\ Slot\ Time] \tag{C.21}$$

where $E[X]$ is the average number of slot times required for successfully transmitting a new frame and is given in [89] by:

$$E[X] = \frac{(1-2p) \times (CW_{min}+1) + p \times CW_{min} \times (1-(2p)^m)}{2 \times (1-2p) \times (1-p)} \tag{C.22}$$

$E[Length\ of\ a\ Slot\ Time]$ is given by:

$$E[Length\ of\ a\ Slot\ Time] = (1-P_{Tx}) \times \sigma + P_{Tx} \times P_s \times T_{success} + P_{Tx} \times (1-P_s) \times T_{collision} \tag{C.23}$$

Appendix D

IEEE 802.11e: Enhancements of the IEEE 802.11 MAC

This section presents IEEE 802.11e and the main enhancements added to the IEEE 802.11 so that QoS is supported. The IEEE 802.11 Task Group E is still in the process of developing and determining the specifications. Here, a general view of the enhancements is given.

D.1 IEEE 802.11 Limitations

Even though the IEEE 802.11 PCF has been designed to support time-bounded services, there are problems that constrain its use. One of them is the unpredictable beacon delay due to the unknown transmission time of the polled stations. At TBTT (see section 3.3), the PC schedules the beacon as the next frame to be transmitted, but it can only be transmitted if the medium has been sensed as idle for at least PIFS. The medium may not be idle at this time and the beacon frame would then be delayed. This would automatically delay the time-bounded data frames that were set to be sent under the CF mode. Another problem is the unknown duration of the data frame transmitted by the polled stations. These frames may have variable lengths and can be sent with different transmission modes. This can not be controlled by the PC [23, 85]. Moreover, data transmitted during CP with DCF is not controlled by the PC and this can delay further time bounded transmissions.

In order to support QoS, the IEEE 802.11 Task Group E is currently developing an enhanced version of the IEEE 802.11 MAC called IEEE 802.11e [40]. This enhanced version defines the Hybrid Coordination Function (HCF) to access the channel. HCF uses the Enhanced DCF (EDCF) also called Enhanced Distributed Channel Access (EDCA) [23] for the contention access period, CP, and the Hybrid Coordination Channel Access (HCCA) for the controlled access and contention free period, CFP. As in the legacy IEEE802.11 MAC, EDCF and HCCA shall alternate in a super-frame and are coordinated by a Hybrid Coordinator (HC). Note that EDCF is not a separate coordination function, but is a part of a single coordination function HCF [38].

D.2 Service Differentiation in EDCF

One major enhancement to support QoS is that the EDCF has the ability to differentiate data using different priorities with multiple queues, whereas the DCF only supported one single FIFO regardless of the priority. It provides a prioritised level of QoS [38, 40, 189]. Each station under EDCF defines 8 user priorities (UP) (also called traffic categories (TC)) and each frame from higher layers is mapped onto one of the four access categories (AC), as shown in figure D.1 [37, 40], by using a specific priority value. Table D.1 defines the mapping from UP to AC, depending on the type of data required for transmission [189, 190].

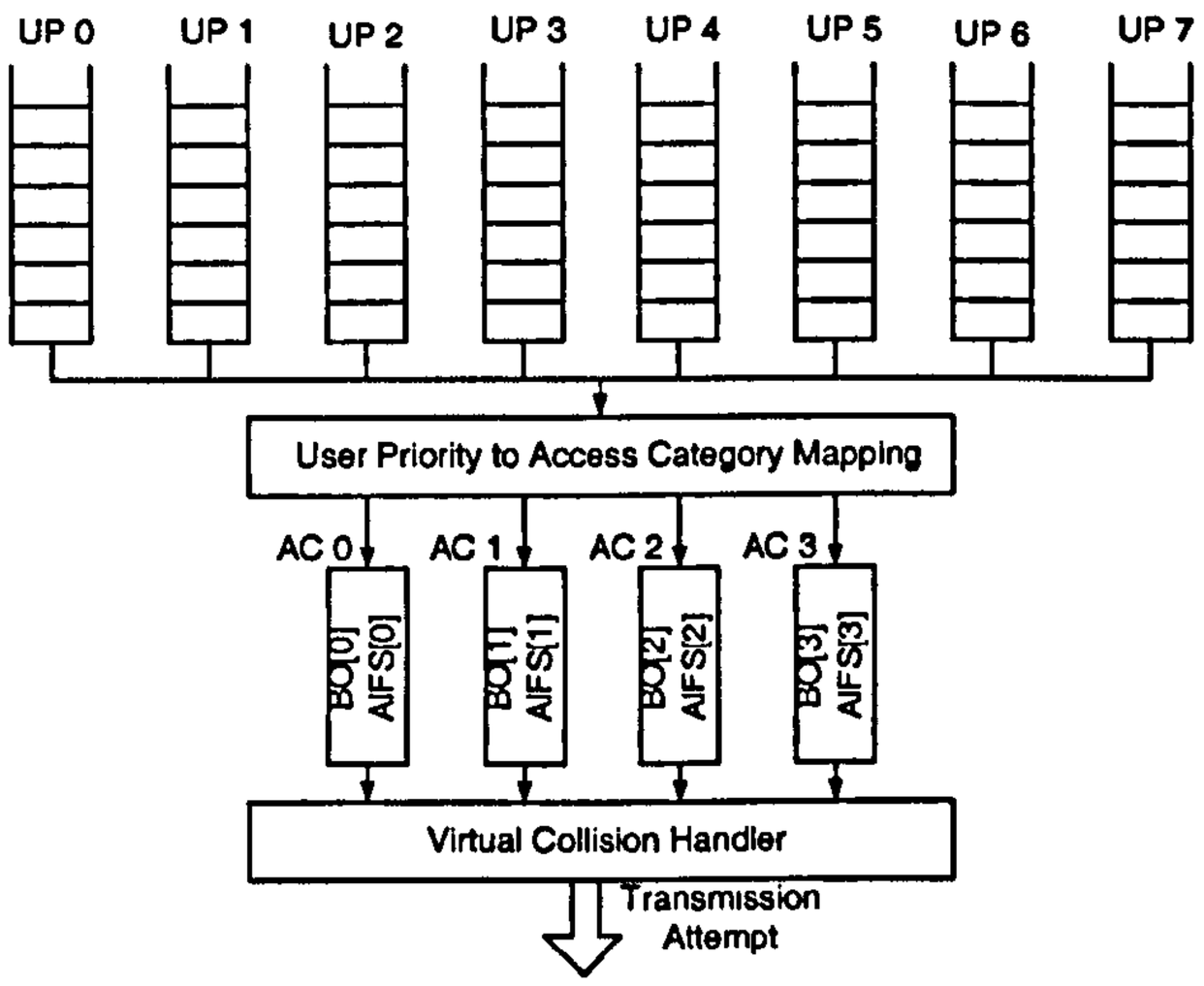


Figure D.1: Four Access Categories for EDCF

Table D.1: User Priority to Access Category Mapping

Priority	*User Priority*	*Access Category*	*Designation*
Lowest	1	0 - AC_BK	BK - Background
	2	0 - AC_BK	BK - Background
	0	1 - AC_BE	BE - Best Effort
	3	1 - AC_BE	EE - Excellent Effort Video
	4	2 - AC_VI	CL - Control Load Video
	5	2 - AC_VI	VI - Video
	6	3 - AC_VO	VO - Voice
Highest	7	3 - AC_VO	NC - Network Control Voice

As shown in figure D.1, a virtual collision handler is used whenever more than one AC finishes the back-off at the same time within one station [23, 191]. The back-off and the contention window required for each packet are parameterised with AC specific parameters. DIFS timing is replaced by an Arbitrary Interframe Spacing (AIFS), which

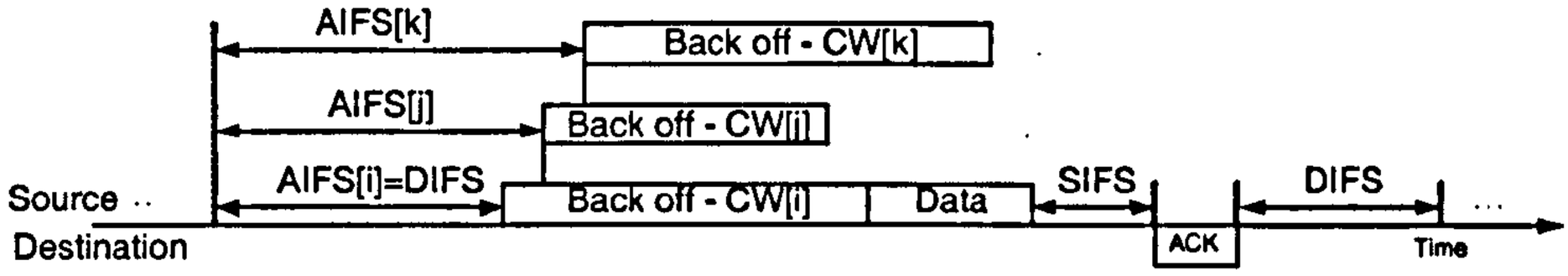


Figure D.2: Multiple back-off Access for EDCF with different categories

is at least as long as DIFS and can be enlarged individually for each AC as shown in figure D.2. CW and AIFS are AC dependent and AIFS is given by:

$$AIFS(AC) = SIFS + \alpha(AC) \times Slot\,Time \tag{D.1}$$

where $\alpha(AC)$ is an integer depending on the AC. The smaller the $\alpha(AC)$) and CW_{min}, the shorter the channel access delay and the higher the priority. Note that PIFS and DIFS would correspond to α equal to 1 and 2 respectively. However, AIFS is at least DIFS, so α is at least 2. With a small AIFS(AC), the waiting time for the medium to be idle is shorter. However, the probability of collision increases when operating with smaller CW_{min} [38]. Table D.2 [190] shows the derivation for typical contention window parameters. Table D.3 shows the AC dependent parameters used for the IEEE 802.11a PHY and under a lightly loaded network. The α values are not defined by the standard but by an external scheduler, carefully chosen depending on the relative priorities to be assigned. $\alpha(AC)$, $CW_{min}(AC)$ and $CW_{max}(AC)$ are announced by the HC in beacon frames.

Table D.2: Typical EDCF Contention Window Parameters for QoS Differentiation

AC	CW_{min}	CW_{max}
0 - AC_BK	CW_{min}	CW_{max}
1 - AC_BE	CW_{min}	CW_{max}
2 - AC_VI	$\frac{CW_{min}+1}{2} - 1$	CW_{min}
3 - AC_VO	$\frac{CW_{min}+1}{4} - 1$	$\frac{CW_{min}+1}{2} - 1$

Table D.3: EDCF Parameters for QoS Differentiation over IEEE802.11a PHY layer

AC	CW_{min}	CW_{max}	α	$AIFS\ \mu s$	$Av.\ Backoff\ \mu s$
0 - AC_BK	15	1023	7	79	67.5
1 - AC_BE	15	1023	5	79	67.5
2 - AC_VI	7	15	4	52	32.5
3 - AC_VO	3	7	2	34	13.5

The achievable throughput for the different access categories are shown in figure D.3 for mode 3. Because of their large IFS compared to DIFS (for the same contention

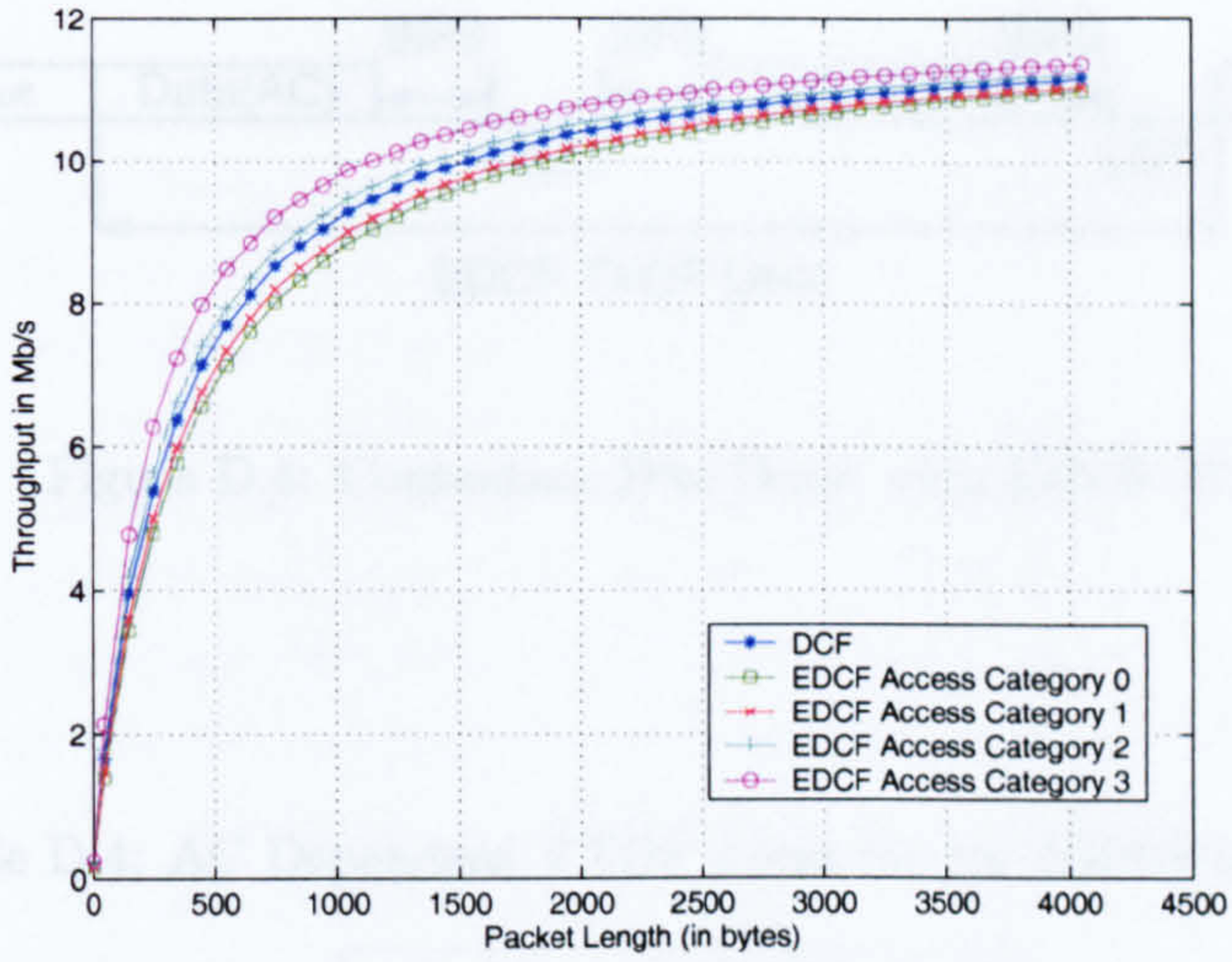


Figure D.3: Throughput for ACs in EDCF, mode 3

window), AC_BK (0) and AC_BE (1) lead to worse performance relative to the legacy DCF scheme. On the other hand, AC_VI (2) and AC_VO (3) offer better throughput performance, even if their AIFS is larger than DIFS (0.8 Mbit/s gain for AC 2 with 500 byte long packets). This is due to their contention windows being much smaller, so stations need to wait for a smaller back-off to access the channel. However, EDCF on its own can not provide effective traffic protection and QoS guarantees. This would only be achievable with a polling scheme [80].

D.3 Transmission Opportunities

IEEE802.11e defines a transmission opportunity (TXOP) as the interval of time when a particular station has the right to initiate transmissions. TXOPs are allocated during the contention period (EDCF TXOP) or granted through the controlled access HCCA (HCCA TXOP) by the HC. The EDCF TXOP is limited by the parameter *TXOP Limit*, which is regularly delivered by the HC in beacon frames along with $\alpha(AC)$, $CW_{min}(AC)$ and $CW_{max}(AC)$ for all the ACs. The HCCA *TXOP limit* is specified by the duration field in poll frames [23, 85]. During a TXOP, a station is allowed to transmit multiple frames from the same AC with a SIFS gap between an ACK and the subsequent frame in the TXOP [37]. This is referred to as Contention Free Burst (CFB). Figure D.4 depicts an EDCF TXOP with 2 frames in the CFB. Each single data frame is acknowledged after a SIFS, and the transmitting station does not need to contend again for the channel.

Note that the *TXOP Limit* is also defined to be AC dependent (and also PHY dependent) and the larger the *TXOP Limit*, the larger the share of capacity for the AC [23]. Table D.4 [40, 192, 193] shows the *TXOP Limit* for the 4 ACs and for the IEEE802.11a PHY Layer. The 802.11e protocol also defines a maximum frame lifetime per AC, which specifies the maximum time a frame can remain in the MAC before it

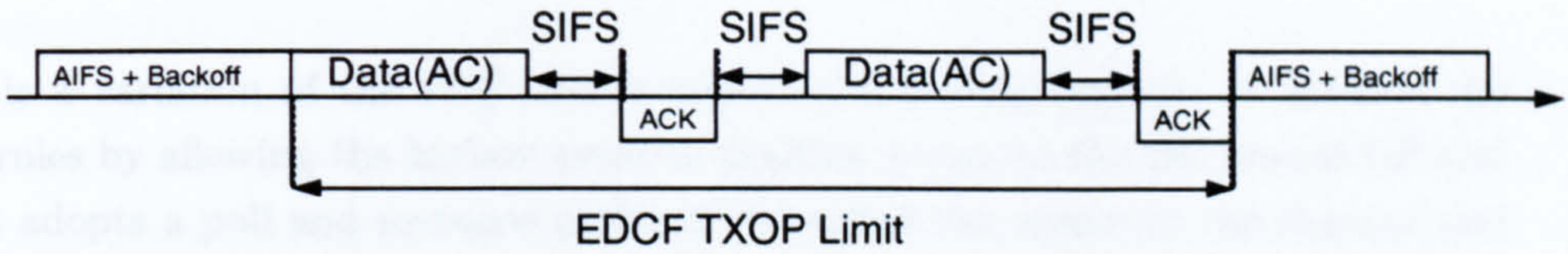


Figure D.4: Contention Free Burst with EDCF TXOP

is dropped.

Table D.4: AC Dependent *TXOP Limit* for the IEEE802.11a PHY

AC	*TXOP Limit*
0 - AC_BK	0 ms
1 - AC_BE	0 ms
2 - AC_VO	3.008 ms
3 - AC_VI	1.504 ms

Each station may ask for a transmission opportunity with its starting time and its duration. These TXOPs improve the throughput performances since overhead are reduced for every transmission [37], as shown figure D.5 for mode 3. With 500 byte long packets, using 10 frames in CFB offers a 1.5Mbits/s gain over the legacy DCF. Throughput improvements are explained by the fact that a station can transmit multiple frames from the same AC consecutively, with a SIFS and an ACK, without contending for the channel as long as the whole transmission time does not exceed the TXOP limit determined by the AP [80]. Overheads are therefore reduced.

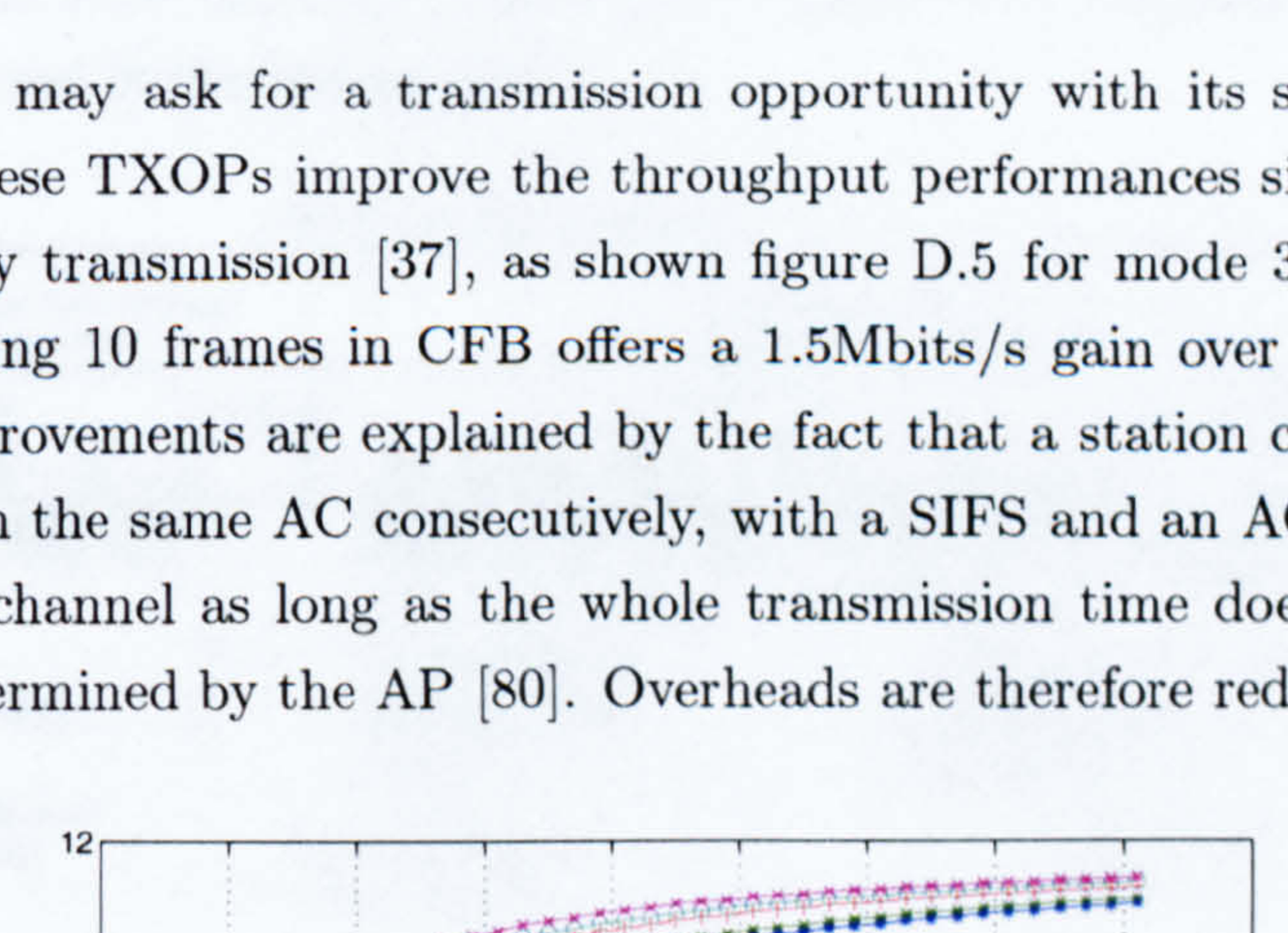


Figure D.5: Throughput with CFB, mode 3

D.4 HCCA: Controlled Access of HCF

HCCA is a variation of the *PCF and* is based on a polling system. It extends the EDCF rules by allowing the highest priority medium access to the HC during *CP and* CFP. It adopts a poll and response protocol to control the access to the channel and makes use of the PIFS, which is shorter than AIFS. Once the HC has control of the medium, it transmits parameterised and contention free CF-poll frames (called QoS CF-poll frames) to stations that have required transmissions [38], as shown in figure D.6 [23, 85]. These QoS CF-poll frames include HCCA TXOP parameters such as the duration granted. Note that, differently from the PCF MAC, HCCA operates during both CFP and CF and the HC can grant TXOP and poll stations during both periods as shown in figure D.6. During CFP, stations that are not polled can not use the channel since their NAV is indicating a busy medium. The HC has free access to the channel and polls stations. During CP, the HC can also use free access to the medium once it becomes idle for at least PIFS in order to deliver data or issue QoS CF-poll frames [38]. Stations require to be polled by sending *Traffic Specification* (TSPEC) elements to the HC. TSPEC contains the set of parameters that characterises the data stations wish to transmit. Once TSPEC is received and processed, the HC schedules the transmission and/or sends a QoS CF-poll frame with QoS requirements specified in TSPEC. Similarly to PCF, the CFP ends with a CF-End frame emitted by the HC or at the time announced in the beacon frame.

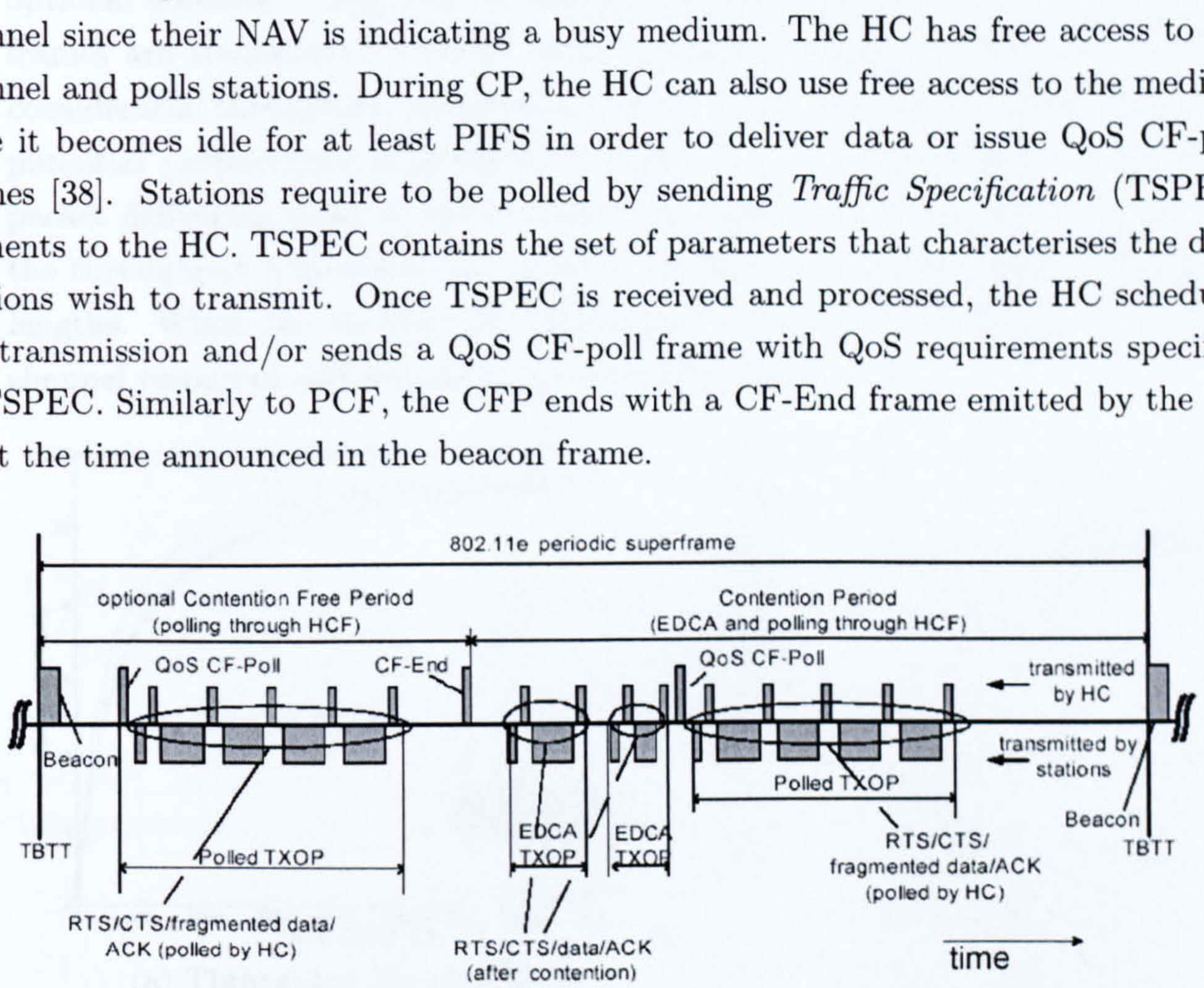


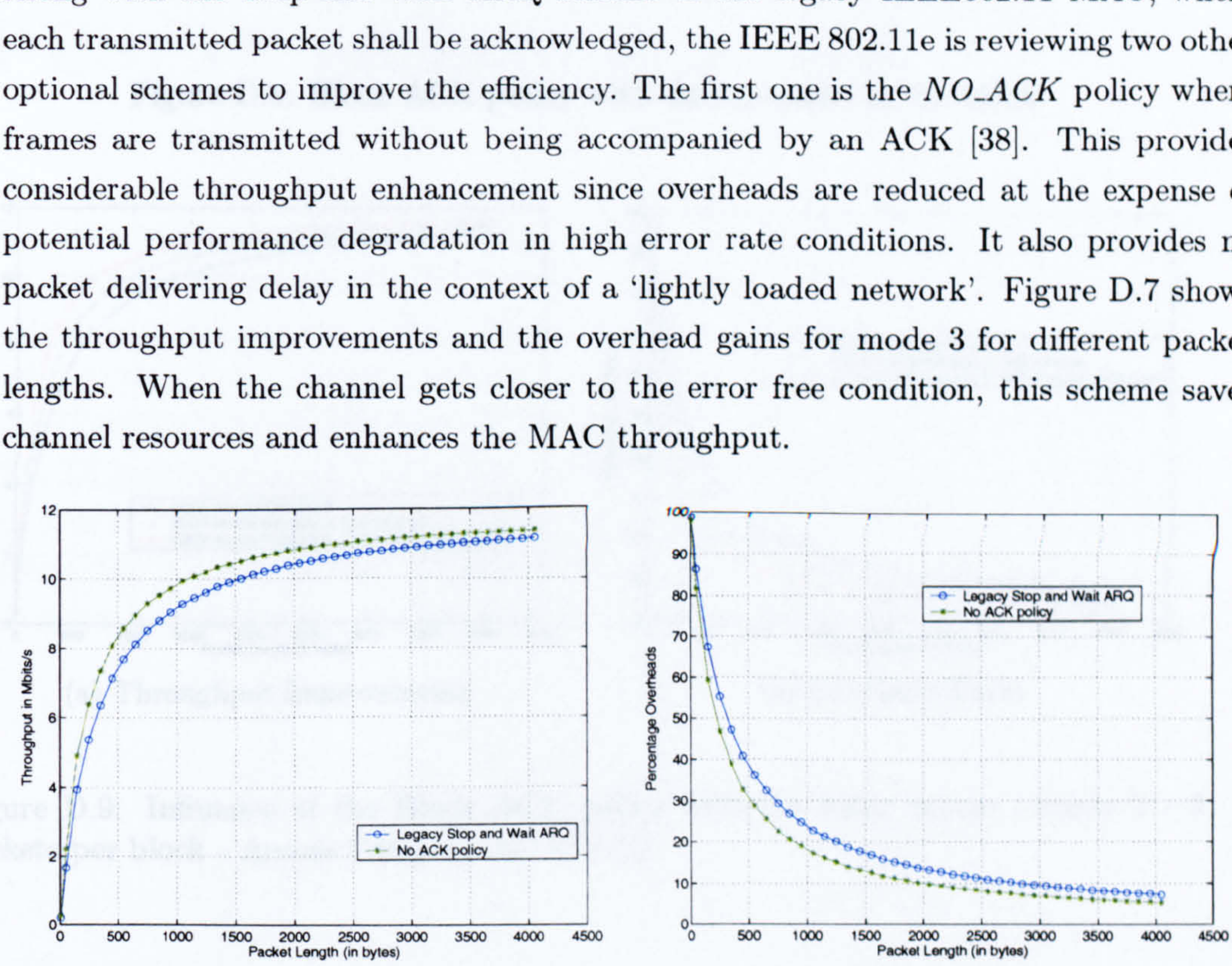
Figure D.6: Channel Access under HCF

The transmission example depicted in figure D.6 is as follows. At TBTT, the HC emits a beacon frame starting the CFP. Only polled stations are allowed to transmit. HC polls one station by sending a QoS CF-Poll frame. The polled station starts the transmission and then follows a RTS/CTS and a Fragment/ACK sequence. The CFP ends when the HC emits the CF-Frame. The CP of the super-frame then starts with the EDCF access. First, one station gains access after a random back-off and uses its EDCF TXOP as specified in the beacon frame to transmit the RTS/CTS/Data/ACK sequence. When the EDCF TXOP has expired, stations contend again until another one gains the channel. A similar exchange sequence is transmitted by the winning

station within its own EDCF TXOP. After one PIFS, the HC emits a QoS CF-poll frame to have control of the channel and poll one station. A HCCA TXOP is then granted to the poll station, which is now able to transmit. The polled transmission in CP finishes at the end of the HCCA TXOP. At TBTT, the CP ends and the HC emits a beacon frame to start a new super frame [23, 38].

D.5 ACK Policies

Along with the Stop-and-Wait ARQ scheme of the legacy IEEE802.11 MAC, where each transmitted packet shall be acknowledged, the IEEE 802.11e is reviewing two other optional schemes to improve the efficiency. The first one is the *NO_ACK* policy where frames are transmitted without being accompanied by an ACK [38]. This provides considerable throughput enhancement since overheads are reduced at the expense of potential performance degradation in high error rate conditions. It also provides no packet delivering delay in the context of a 'lightly loaded network'. Figure D.7 shows the throughput improvements and the overhead gains for mode 3 for different packet lengths. When the channel gets closer to the error free condition, this scheme saves channel resources and enhances the MAC throughput.



(a) Throughput Improvements (b) Overhead Gains

Figure D.7: Influence of the No ACK policy with the basic access, mode 3

Another optional ACK feature, called *Block ACK*, allows a station to transmit a number of packets consecutively during one TXOP without individual ACK frames. Packets transmitted during the TXOP are referred to as a block [23]. At the end of the block, all the packets are acknowledged together with a single ACK frame, where a bit pattern specifies which packets in the blocks need to be re-transmitted. Figure D.8 shows an example where a block contains three packets. Figure D.9 shows the improvements in terms of throughput and overhead for mode 3, with AC_VO (2) and three frames per block. For a packet length of 500 bytes, the standard EDCF offers a throughput of 7.9 Mbits/s and a 34% overhead. The use of CFB with three packets decreases the overhead to 26% and increases the throughput to 8.8 Mbits/s. The use of CFB with three packets and the *Block ACK* policy further decreases the overhead to

20% and the throughput is enhanced to 9.5Mbits/s. When fragmentation is used and when fragments are corrupted, this policy allows the retransmission of the erroneous fragments only, without the need of retransmitting all the blocks.

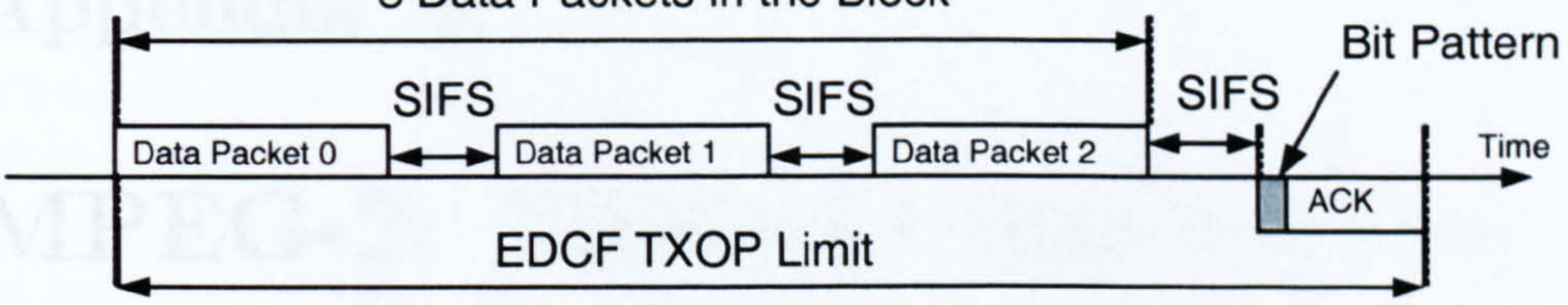
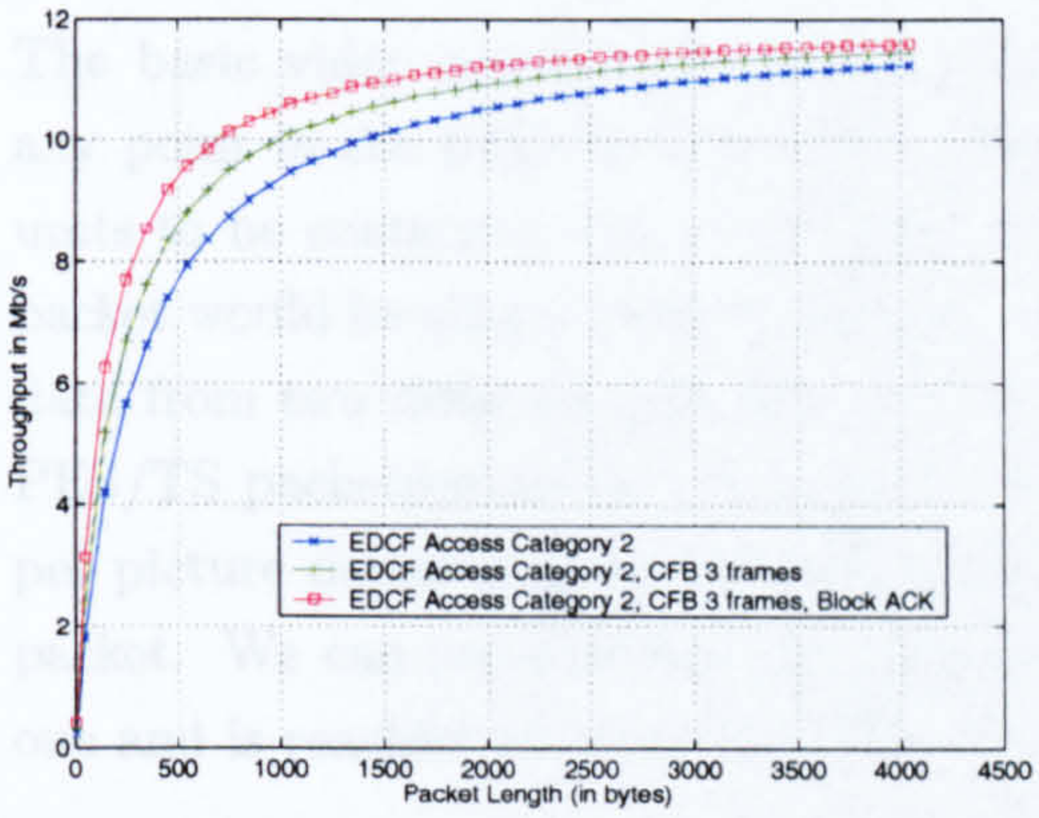
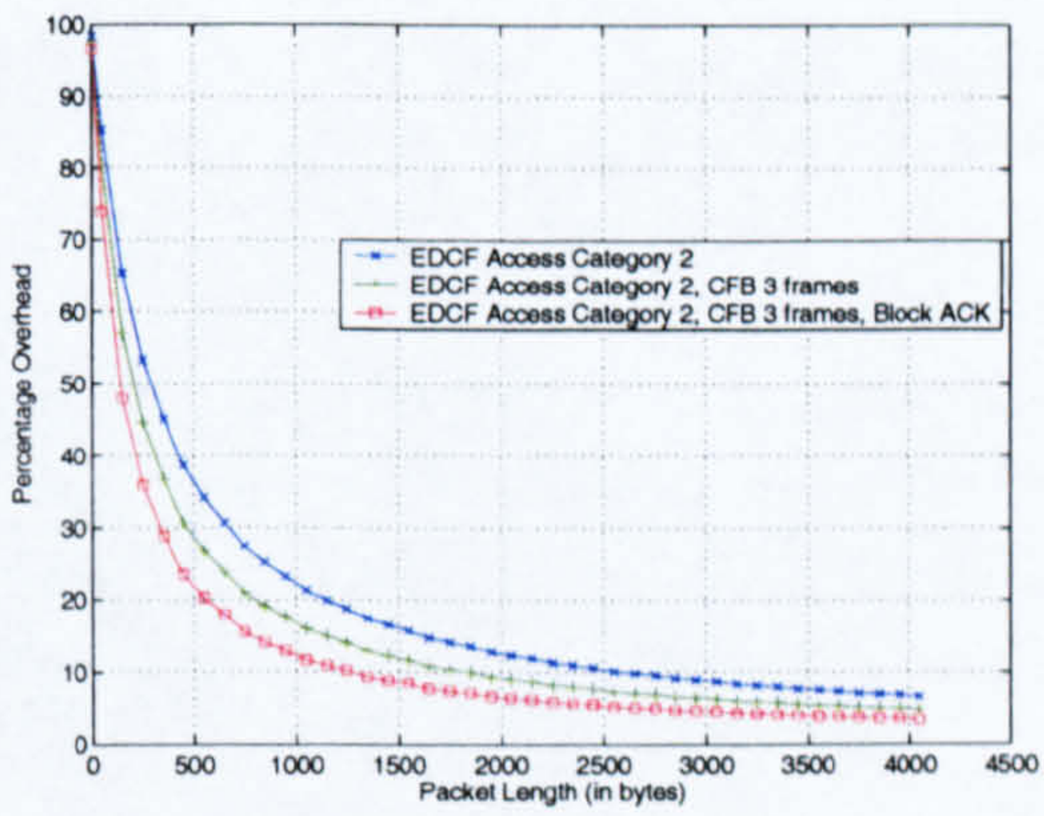


Figure D.8: Block ACK policy with three packets in the block



(a) Throughput Improvements



(b) Overhead Gains

Figure D.9: Influence of the Block ACK policy with the basic access - mode 3 - 3 packets per block - Access Category AC_VO (2)

Appendix E

MPEG-2: Video Access unit to PES/TS packets mapping

The basic video access unit considered is a slice. A video access unit may start at any point in the payload of a PES packet. It is also possible for several small access units to be contained in a single PES-packet. Nevertheless it is assumed that a PES-packet would be aligned with a video start code prefix and a PES packet cannot contain data from two different pictures. This assumes that data encryption occurs after the PES/TS packetisation and not before. Figure E.1 shows the number of PES packets per picture depending on the number of video access units (slices) included in a PES packet. We can see that the minimum number of PES packets per frame is equal to one and is reached when all the slices of a frame are included in the PES packet.

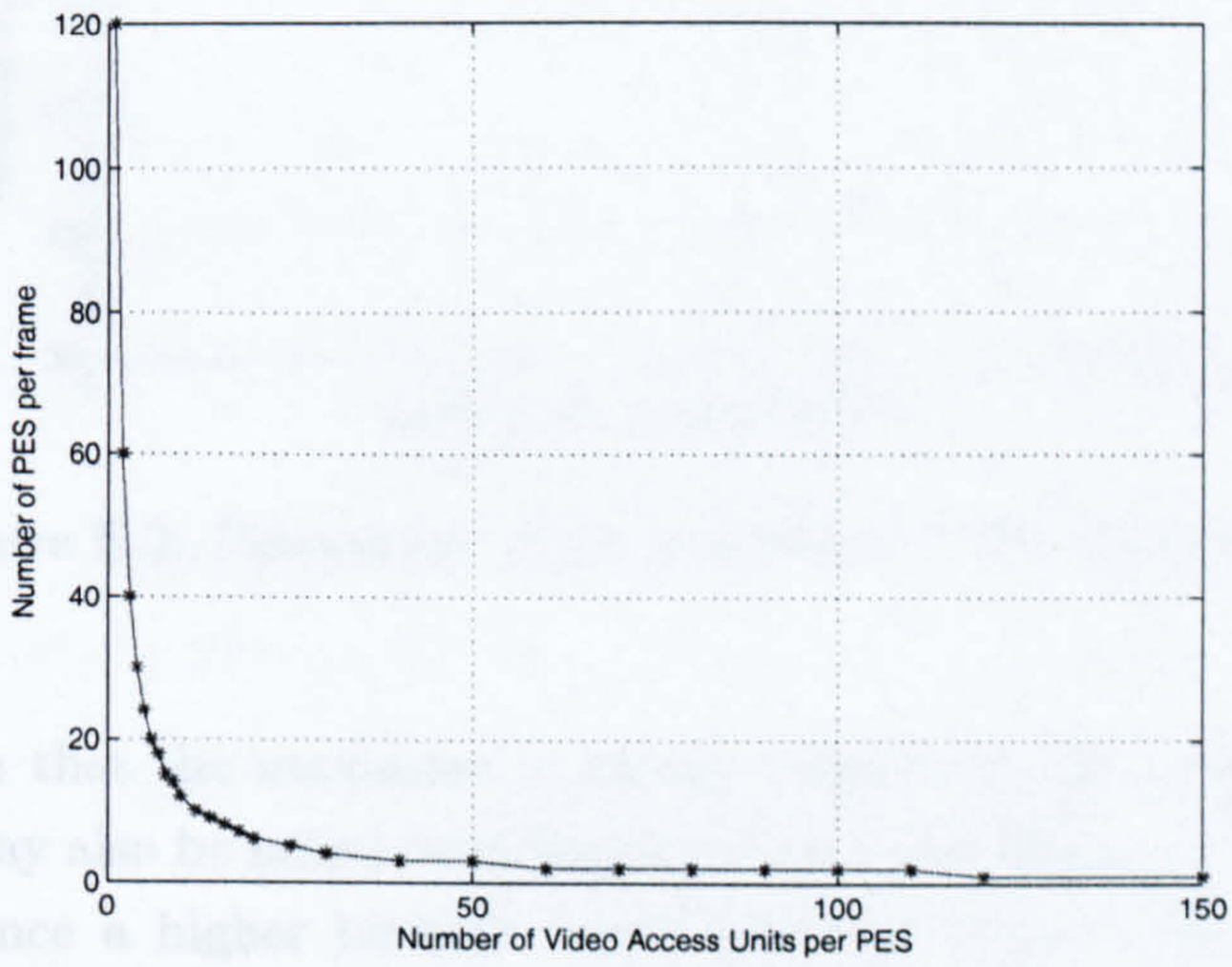


Figure E.1: PES repartition per frame

Depending on the number of PES packets per frame, the overhead introduced by the PES packet header will vary. The adaptation field control (AFC) field will also introduce extra overhead. Having more TS packets per PES packet will decrease the impact of the AFC overhead since all the TS packets minus one will be full payload. On

the other hand if there are fewer TS packets in a PES packet, the proportion of AFC overhead will increase. Figure E.2 shows the percentage of video data in the output TS stream after the video packetisation versus the number of video access units per PES-packet. It can be seen that by mapping only one video access unit per PES packet, the resulting packetised TS stream will contain a large amount of overhead due to 4 bytes of TS header, 9 bytes of PES header and a variable length AFC overhead (only 33% of data for a video encoded at 2Mbit/s and 55% of data at 4Mbit/s). This is especially the case where a very small access unit (a couple of dozen bytes) will be mapped onto 1 PES packet and will lead to only one TS packet. The TS packet will therefore contain the PES header and the AFC field. As the number of video access units increases in the PES-packet, the PES packet length increases and therefore generates more TS packets. The efficiency is then improved. The theoretical maximum achievable would be 184/188 = 97.87%, where the amount of PES-packet headers and the AFC length would be small enough to be neglected and just the 4 byte TS header is taken into account.

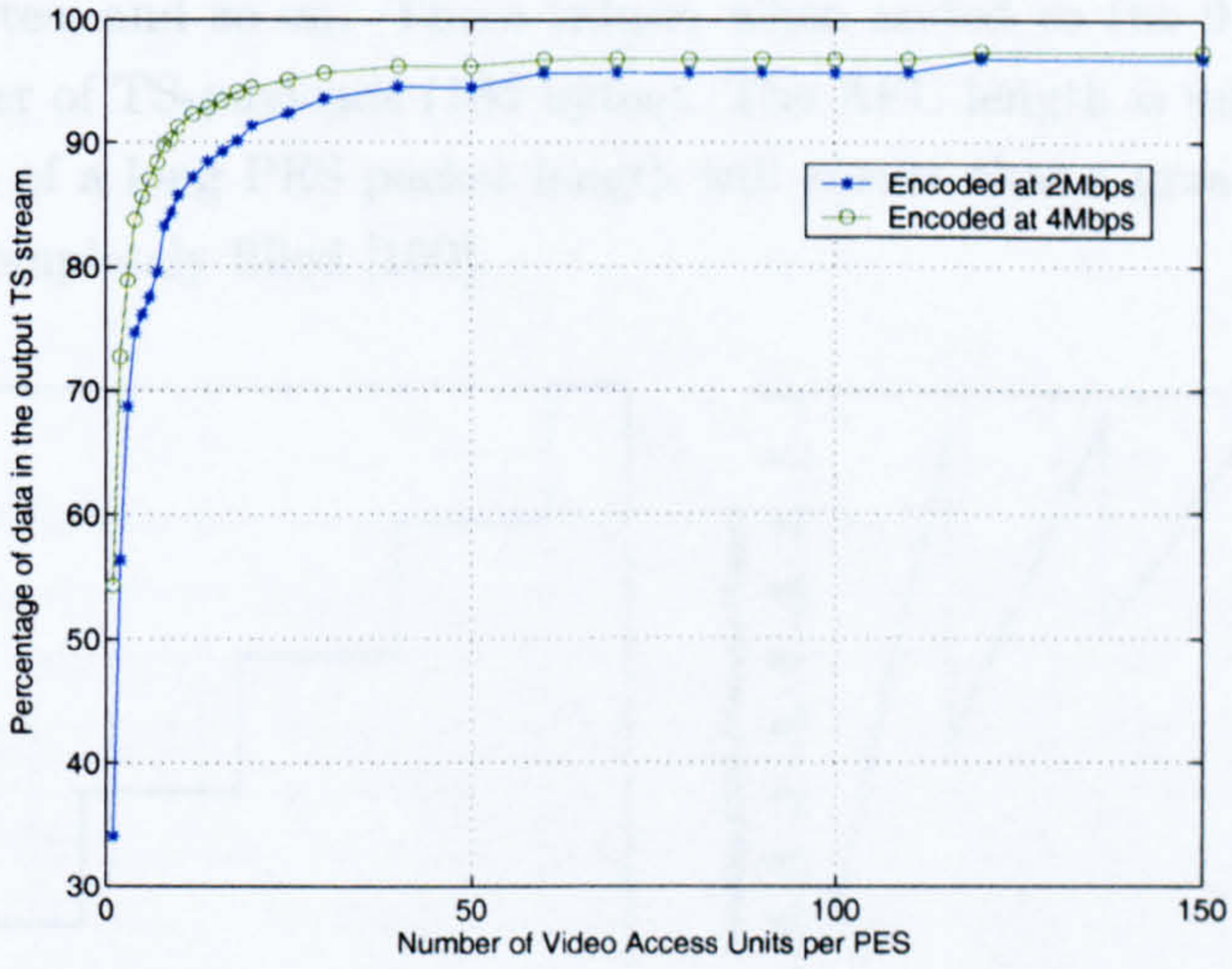


Figure E.2: Percentage of video payload in the TS-stream

It can be seen that the maximum is almost reached for 50 video access units per PES packet. It may also be noted from figure E.2 that the efficiency differs for different video bit-rates since a higher bit-rate would generate longer video access units. It therefore leads to a larger number of TS packet per PES packet (see figure E.3) and reduces the amount of PES-headers and AFC fields.

Another option would be to do the packetisation with a fixed PES packet length, irrespective of the video access unit boundaries, so that the length would an integer number of TS packets. This would minimise the effect of AFC. The choice of the length would determine the number of TS packets in a PES packet as shown in igure E.4(a). It therefore determines critical values where the AFC length would be zero and where the efficiency would be maximal. This is shown in Figure E.4(b).

The critical values for PES packet payload are 175 bytes, 359 bytes, 543 bytes, 727

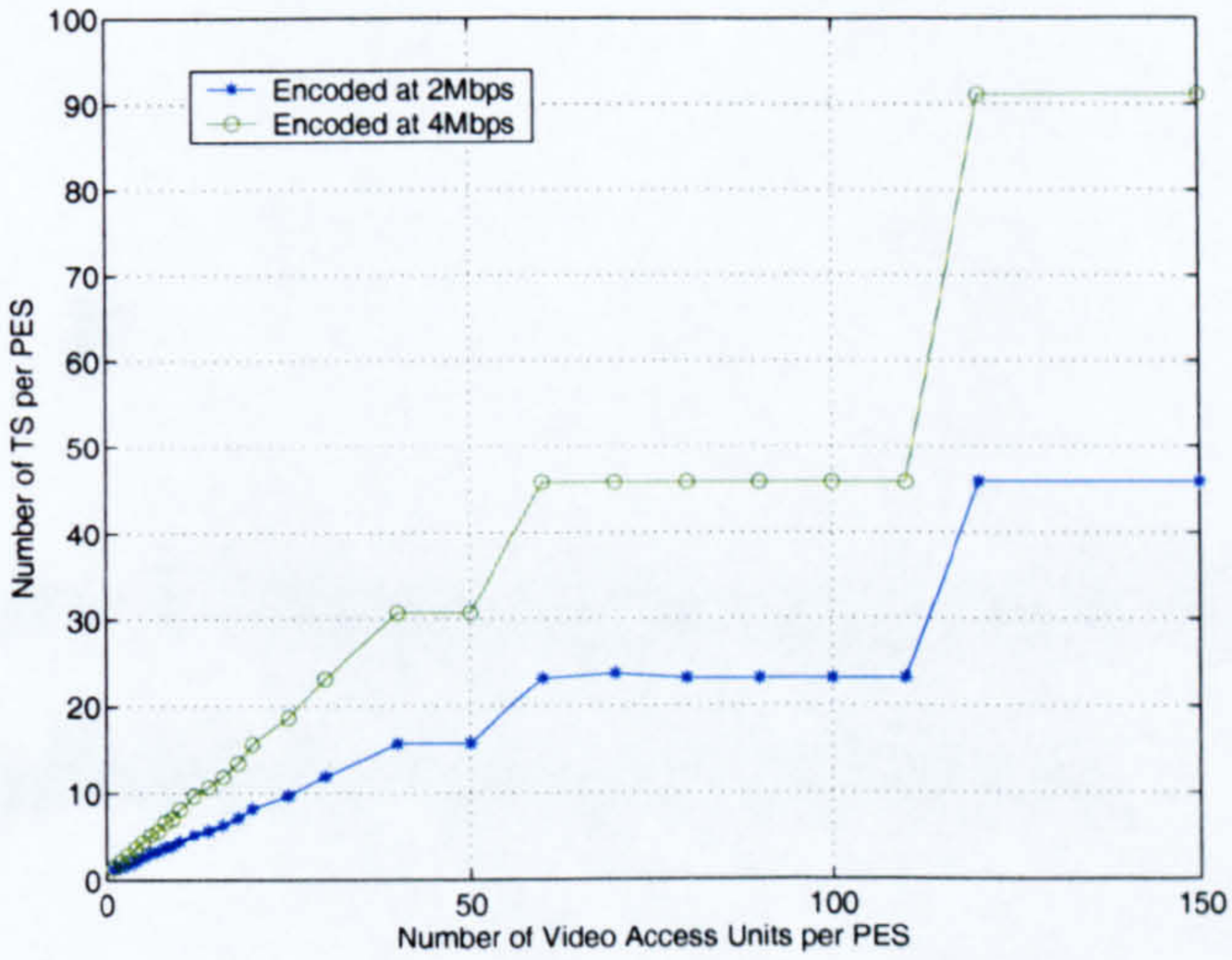
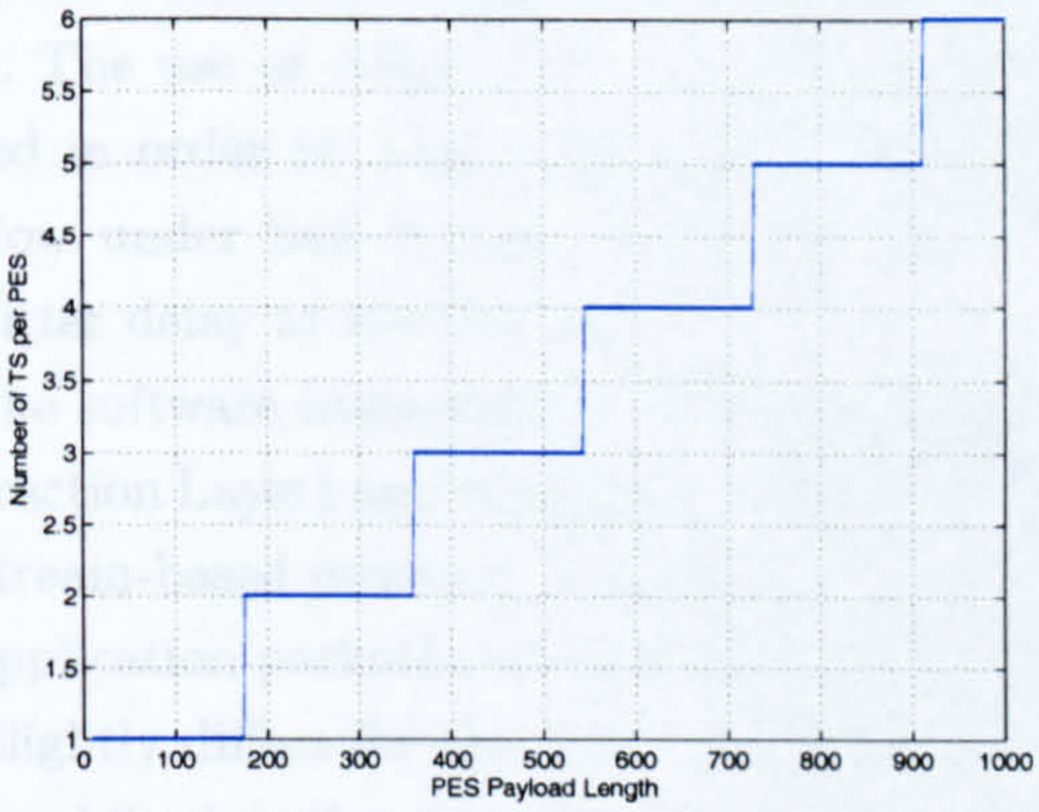
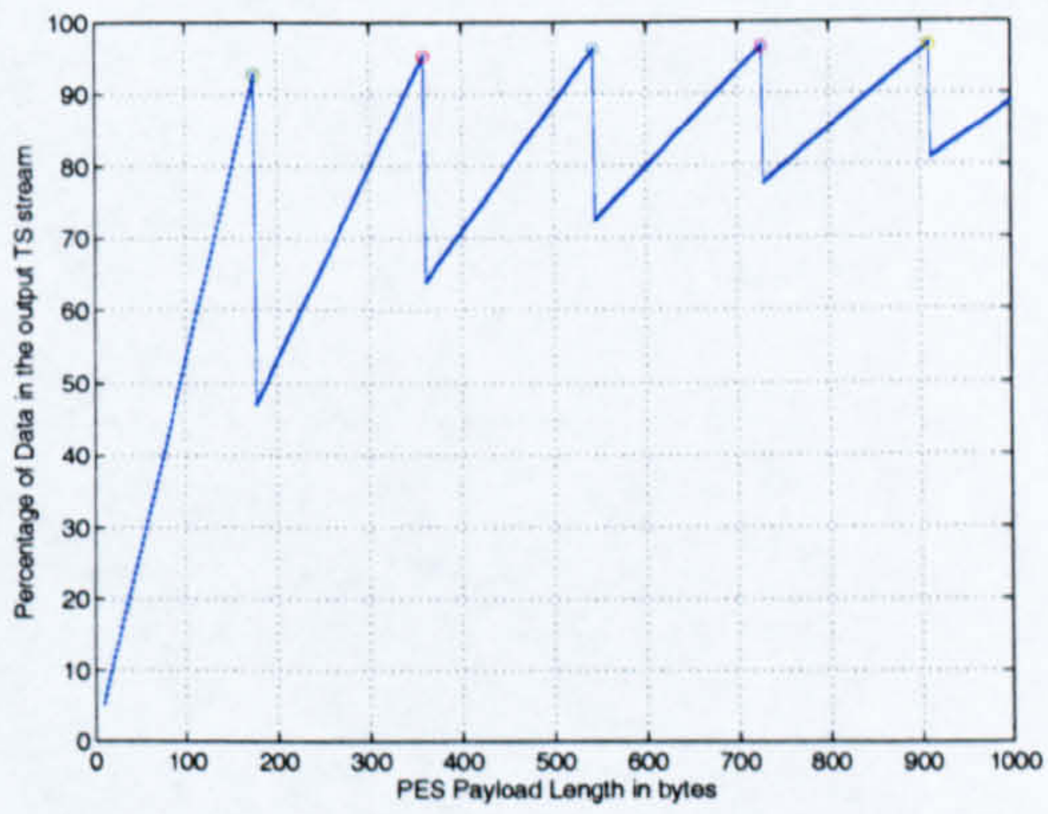


Figure E.3: Number of TS-packets per PES-packet

bytes, 911 bytes, and so on. These values, when added to the 9 byte header, are an integer number of TS-payloads (184 bytes). The AFC length is minimum and equal to 0. The choice of a long PES packet length will ensure that a greater proportion of TS packets are completely filled [100].



(a) Number of TS per PES



(b) Percentage of video payload

Figure E.4: Influence of fixed PES-payload length

Appendix F

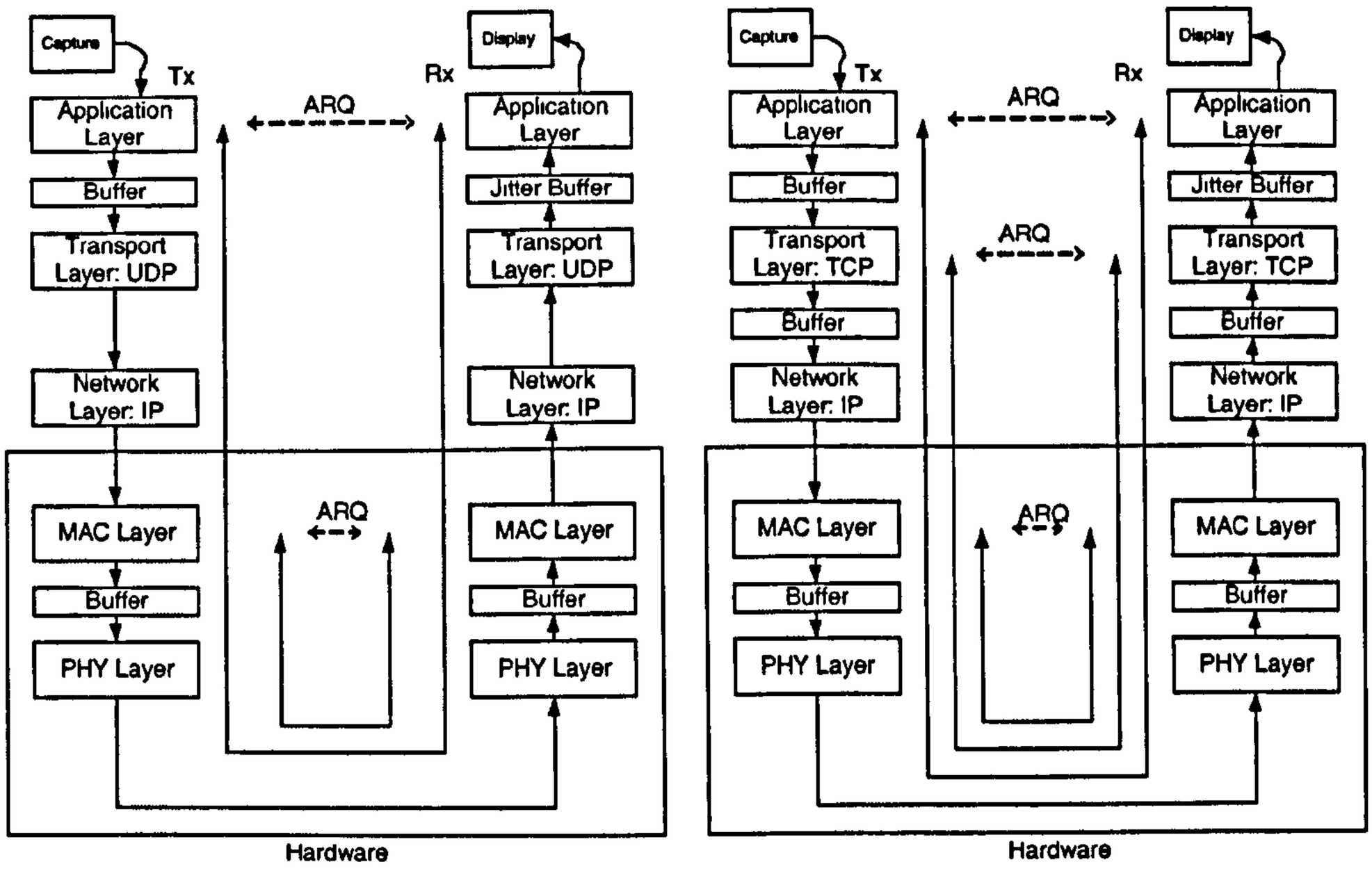
ProVision Communications Cross Layer Software Description

The client/server software pair used in chapter 8 was developed by Tuan Kiang Chiew from ProVision Communication Technologies Ltd [182] with Visual Studio 6.0^{TM}. Both client and server are using either the UDP/IP stack or the TCP/IP stack as shown in figure F.1. UDP, TCP and IP were briefly described in section 6. Both TCP/IP and UDP/IP functionalities have been implemented with *Microsoft*^{TM} Winsock32 API Version 2. For more details on the implementation of the protocol stack, please refer to [182]. Note that in this protocol stack, ARQ is present at the MAC layer requiring therefore a buffer, for both TCP and UDP cases. TCP implements another level of ARQ. The use of ARQ at the application layer is up to the application. A buffer is needed in order to control the transmission rate at the encoder and also to prevent overflow under bad channel conditions, and in order to correct out-of-order packets and jitter delay at the decoder.

The software implements a RTP-like layer between the application layer (Network Abstraction Layer) and the UDP or TCP layers with a 16 byte header. Because TCP/IP is a stream-based protocol, it implements its own intrinsic packetisation, which removes the application packetisation structure based on NAL units. This RTP-like layer therefore slightly differs for the UDP and TCP cases [182]. One laptop implements a static server while the other implements a static or mobile client used to collect transmission statistics in the form of log files. The inputs to the software include:

- Packet Size at the application level (before UDP/IP or TCP/IP packetisation)
- Link speed (fixed mode or automatic Link Adaptation). The link speed is manually set with the driver toolbox at the server side to one specific mode or with the automatic Link Adaptation algorithm of the chipset.
- Video transmission target rate, or as fast as possible.
- The choice between TCP, UDP-Unicast and UDP-Broadcast (with no MAC ARQ)

Note that the IEEE 802.11a *CISCO*^{TM} card allows to the maximum number of



(a) UDP (b) TCP

Figure F.1: Protocol Stack with UDP and TCP

retries at the MAC to be altered. The IEEE 802.11b/g $Belkin^{TM}$ and $Buffalo^{TM}$cards do not have such flexibility. The software logs the following data:

- On a time basis (per second) (see sample in figure F.2(a))
  - Time in seconds
  - The target transmission bit rate in kbits/s
  - Average packet delay in ms
  - Jitter computed as the standard deviation of the delay in ms
  - Jitter computed as the Max-Min of the delay in ms
  - PER during a one second window
  - Instantaneous Received Signal Strength Information (RSSI) at the receiver in dBm (at the time of sampling)
  - Instantaneous link speed (at the time of sampling) from the server in Mbits/s
- On a packet basis (see sample in figure F.2(b))
  - Packet arrival time in ms
  - Packet counter (running number in the RTP-like packet header).
  - Packet Size
  - Packet delay in ms

- Offset of the total number of bytes sent (with an offset of 16 bytes per packet due to the RTP-like header)
- RSSI of the receiver in dBm
- The link speed from the server in Mbits/s

I UDP(150)@200kbps Video=0. Starts: Thu Jul 22 11:43:26 2004

t(sec)	kbps	delay	jitter1	jitter2	PER	RSSI	LinkSpeed
1.0	196.23	23.44	2.727	28.0	0.0000	-80	24
2.0	200.20	23.92	0.963	5.0	0 0000	-80	24
3.0	196.39	38.94	27.695	107.0	0 0181	-80	24
4.0	191 86	24.94	4.333	30.0	0 0417	-80	24
5.0	197.59	25.20	6.123	47.0	0.0120	-73	24
6.0	196.19	32.57	21.586	100.0	0.0181	-75	24
7.1	165.70	53.96	58.045	217.0	0.0427	-80	24
8.0	216.06	55.71	44.404	155.0	0.0765	-80	24
9.0	198.60	141 96	127.871	423.0	0.0119	-75	24
10.2	143.33	37.58	34.155	260.0	0.0743	-82	24

(a) Per second log

I UDP(150)@200kbps Video=0. Starts: Thu Jul 22 11:43.26 2004

t(msec)	pkt_counter	size	delay	offset_from_start	RSSI	LinkSpd
0	00000000	150	0	0000000000000000	-78	24
0	00000001	150	0	0000000000000134	-78	24
35	00000002	150	23	0000000000000268	-78	24
36	00000003	150	24	0000000000000402	-78	24
48	00000004	150	24	0000000000000536	-80	24
48	00000005	150	24	0000000000000670	-80	24
60	00000006	150	24	0000000000000804	-80	24
60	00000007	150	24	0000000000000938	-80	24
71	00000008	150	23	0000000000001072	-80	24
72	00000009	150	24	0000000000001206	-80	24

(b) Per packet log

Figure F.2: Samples of log files

PER and delays are taken at the RTP-like layer. RSSI and link Speed are taken at the PHY layer. The software allows therefore a cross-layer analysis of the transmission for both UDP and TCP transmissions. The packet counter allows us to determine which packets are missing (a packet is missing if there is a gap between consecutive packet counter values). Note that with TCP, packets can not be out of sequence. Moreover, in a single hop network, UDP packets can not either be out of sequence. In addition, one packet can be duplicated when the packet is received correctly but the MAC ACK is received corrupted. The MAC therefore attempts another transmission. Error patterns at the RTP-like layer have been generated from the log of the packet counter (0 meaning packet received correctly and 1 meaning missing packet).

Bibliography

[1] Mohammed E.Al-Mualla, C.Nishan Canagarajah, and David R.Bull. *Video Coding for Mobile Communications: Efficiency, Complexity and Resilience.* Academic Press - Elsevier Science, 2002.

[2] IEEE Std 802.11b-1999; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer extension in the 2.4GHz Band, 1999.

[3] IEEE Std 802.11g; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further High-Speed Physical Layer in the 2.4GHz Band, d1.1, 2001.

[4] IEEE Std 802.11a; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5GHz Band, d7.0, 1999.

[5] Broad Radio Access Network (BRAN); HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional Specification), v 1.2.1, ETSI EN 300 652, July 1998.

[6] ETSI Hiperlan Type 2; Broadband Radio Access Network (BRAN); HIPERLAN Type 2; Physical (PHY) Layer, ETSI TS 101 475 v1.1.1., April 2000.

[7] Andrew R.Nix, Cengiz Evci Mark A.Beach, M.Umehira, and Masaharu Araki. *Insight into Mobile Multimedia Communication*, chapter 36: High-performance Wireless LAN Developmetns for Future Multimedia Communication. Academic Press, 1999.

[8] C.l'Anson, H. Tominaga, T. Wilkinson, M.Yabusaki, D.RBull, A.R.nix, and C.N.Canagarajah. *Insight into Mobile Multimedia Communication*, chapter 1: Mobile Multimedia Communcations - Research Trends and Technical Developments. Academic Press, 1999.

[9] J. Postel. Request For Comments - RFC 791 Internet Protocol, September 1981.

[10] John G. Apstolopoulos, Wai-Tina Tan, and Susie J. Wee. Video Streaming: Concepts, Algorithm and Systems. Technical report, Hewlett-Packard - Molbile and Media Systems Laboratory, 2002.

[11] Yao Wang, Stephan Wenger, Jiangtao Wen, and Aggelos K.Katsaggelos. Error Resilient Video Coding Techniques - Real-Time Video Communications over Unreliable Networks. In *IEEE Signal Processing Magazine*, July 2000.

[12] James Chung-How and David Bull. Loss Resilient H.263+ Video over the Internet. In *Elsevier Journal on Signal Processing: Image Communication*, volume 16, 2001.

[13] L. Doyle, A. Kokaram, and D.O'Mahony. Error-Resilience in Multimedia Applications over Ad-hoc. In *IEEE International Conference On Accoustics, Speech and Signal Processing (ICASSP)*, volume 3, Salt Lake City, May 2001.

[14] S.Rane, A. Aaron, and B. Girod. Systematic Lossy Forward Error Protection for Error Resilient Digital Video Broadcasting. In *Visual Communications and Image Processing Conference (VCIP)*, San Jose, January 2004.

[15] S.Belfiore, M.Grangetto, E.Magli, and G.Olmo. An Error Concealment Algortihm for Streaming Video. In *IEEE International Conference on Image Processing (ICIP)*, Barcelona, September 2003.

[16] Ye-Kui Wang, Miska M. Hannuksela, Viktor Varsa, Ari Hourunranta, and Moncef Gabbouj. The Error Concealment Feature in the H.26L Test Model. In *IEEE International Conference on Image Processing (ICIP)*, Rochester, September 2002.

[17] Emmanuel Jammeh and Mohammed Ghanbari. Transmission of Pre-encoded Video over Best-Effort IP Networks. In *IEEE Packet Video workshop (PV)*, Nantes, April 2003.

[18] Performance Analysis for Video Streams across Networks. Technical report, Net-Predict Inc, December 2003.

[19] Victor S. Frost. Quantifying the Temporal Characteristics of Network Congestion Events for Multimedia Services. In *IEEE Transactions on Multimedia*, volume 5, September 2003.

[20] Cheng Huang, Philip Chou, and Anders Klemts. Optimal Coding Rate Control for Scalable Streaming Media. In *IEEE Packet Video workshop (PV)*, Irvine, December 2004.

[21] Bernd Girod and Niko Farber. Feedback-Based Error Control for Mobile Video Transmission. In *Proceedings of the IEEE*, volume 87, October 1999.

[22] Daji Qiao and Sunghyun Choi. Goodput Analyis and Link Adaptation for IEEE 802.11a Wireless LANs. In *IEEE Transactions on Mobile Computing*, volume 1, October 2002.

[23] Stefan Mangold, Sunghyun Choi, Guido R. Hiertz, Ole Klein, and Bernhard Walke. Analysis of IEEE802.11e for QoS Support in Wireless LANs. In *IEEE Wireless Communications*, December 2003.

[24] Daji Qiao and Sunghyun Choi. Goodput Enhancement of IEEE802.11a Wireless LAN via Link Adaptation. In *IEEE International Conference on Communications (ICC)*, Helsinki, June 2001.

[25] Guiseppe Bianchi. Performance Analysis of the IEEE802.11 Distributed Coordination Function. In *IEEE Journal on Selected Areas in Communications*, March 2000.

[26] Angela Doufexi, Simon Armour, Michael Butler, Andrew Nix, and David Bull. A Study of the Performance of Hiperlan/2 and IEEE 802.11a Physical Layers. In *IEEE Vehicular Technology Conference (VTC)*, Rhodes, Spring 2001.

[27] Angela Doufexi. *Performance of OFDM based Multimedia Wireless Local Area Networks, with and without Antenna Sectorisation.* PhD thesis, University of Bristol, January 2002.

[28] Magis Networks. orthogonal Frequency Division Multiplexing OFDM explained. `http://www.magisnetworks.com`, February 2001.

[29] P. Shelswell. The COFDM Modulation system: The Heart of Digital Audio Broadcasting. Technical report, BBC, 1996.

[30] Stephan Wenger. H.264/AVC over IP. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 13, July 2003.

[31] Hrvoje Jenkac, Thomas Stockhammer, and Gabriel Kuhn. On Video Streaming over Variable Bit-Rate and Wireless Channels. In *IEEE Packet Video workshop (PV)*, Nantes, April 2003.

[32] ETSI DVB-T; Digital Video Broadcasting (DVB), Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television (DVB-T), ETSI EN 300 744, March 1997.

[33] ETSI DVB-H; Digital Video Broadcasting (DVB), Transmission for Handheld Terminals DVB H, A081, ETSI EN 302 304, November 2001.

[34] Thomas Stockhammer, Miska M.Hannuskela, and Thomas Wiegand. H.264/AVC in Wireless Environments. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 13, July 2003.

[35] Thomas Wiegand, Gary J.Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the H.264/AVC Video Coding Standard. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 13, July 2003.

[36] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. ITU-T H.264 - Series H: Audiovisual and Multimedia Systems - Advanced Video Coding for Generic Audiovisual Services, March 2005.

[37] Sunghyun Choi, Javier Del Prado, Sai Shankar, and Stefan Mangold. IEEE802.11e Contention-Based Channel Access (EDCF) Performance Evaluation. In *International Conference on Communications (ICC)*, Anchorage, May 2003.

[38] Javier del Prado Pavòn and Sai Shankar. Impact of Frame Size, Number of Stations and Mobility on the Throuhgput Performance of the IEEE 802.11e. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, March 2004.

[39] IEEE Std 802.11; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[40] IEEE Std 802.11e; Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification: Medium Acess Control (MAC) Enhancements for Quality of Service (QoS), November 2002.

[41] Mobile Communications Departement. Research Activity Report 2004. Technical report, Eurecom - Sophia Antipolis, 2004.

[42] T. Matsumoto, J. Ylitalo, and M. Juntti. Overview and Recent Challenges Towards Multiple-Input Multiple Output Communications Systems. In *IEEE Transactions on Vehicular Technology*, volume 50, May 2003.

[43] R.K Mallik, M.Z Win, M-S Alouini J.W Wao an J.W Shao, and A.J Goldsmith. Channel Capacity of Adaptive Transmission with Maximal Ration Combininb in Correlated Rayleigh Channel. In *IEEE Transactions on Wireless Communications*, volume 3, July 2004.

[44] Angela Doufexi, Simon Armour, Peter Karlsonn, Michael Butler, Andrew Nix, and David Bull. A Comparison of the Hiperlan/2 and IEEE802.11a Wireless LAN Standards. In *IEEE Communication Magazine*, May 2002.

[45] Angela Doufexi, Andrew Nix, and Mark Beach. Combined Spatial Multiplexing and STBC to Provide Throughput Enhancements to Next Generation WLANSs. In *Mobile and Wireless Communications Summit (IST - MWC)*, Dresden, June 2005.

[46] Angela Doufexi, Eustace Tameh, Andrew Nix, Simon Armour, and Aracelo Molina. Hotspot Wireless LANs to Enhance the Performance of 3g and Beyond Cellular Networks. In *IEEE Communication Magazine*, July 2003.

[47] Angela Doufexi, James Chung-How, Andrew Nix, and David Bull. Application of Sectorised Antennas and STBC to Increase the Capacity of Hot Spot WLANs in

an Interworked WLAN/3G Network. In *IEEE Vehicular Technology Conference (VTC)*, Milan, May 2004.

[48] Mathieu Lacage, Hossein Manshaei, and Thierry Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. Technical report, Institut National de Recherche en Informatique et en Automatique (INRIA), May 2004.

[49] Fethi Filali. Dynamic and Efficient Tuning of IEEE 802.11 for Multimedia Applications. In *Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Barcelona, September 2004.

[50] Mihaela van der Schaar, Santhana Krishnamachari, Sunghyun Choi, and Xiaofeng Xu. Adaptive Cross-Layer Protection Strategies for Robust Scalable Video Transmission Over WLANs. In *IEEE Journal on Selected Areas in Communications*, volume 21, December 2003.

[51] E.Settom, X.Zhu, and B.Girod. Minimising Distortion for Multipath Video Streaming over ad hoc Networks. In *IEEE International Conference on Image Processing (ICIP)*, Singapore, October 2001.

[52] T.Yoo, E.Settom, X.Zhu, A.Goldsmith, and B.Girod. Video Streaming Over a Cross-layer Designed ad-hoc wireless network. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, Siena, October 2004.

[53] Bernd Girod, Mark Kalman, Yi.J Liang, and Rui Zhang. Advances in Channel-Adaptive Video Streaming. In Wiley, editor, *Journal of Wireless Communications and Mobile Computing*, volume 2, 2002.

[54] Mark Kalman and Bernd Girod. Techniques for Improved Rate-Distortion Optimised Video Streaming, in print.

[55] Eric Setton, Taesang Yoo, Xiaqing Zhu, Andrea Goldsmith, and Bernd Girod. Cross-Layer Design of Ad-Hoc Networks for Real-Time Video Streaming. In *IEEE Wireless Communications*, August 2005.

[56] Xiaoqing Zhu, Eric Setton, and Bernd Girod. Congestion-Distortion Optimised Video Transmission Over Ad-Hoc Networks. In *European Association for Signal Processing (EURASIP), Video/Image Processing and Multimedia Communications*, 2005.

[57] Tim Wilkinson. *Insight into Mobile Multimedia Communication*, chapter 37: HIPERLAN - An Air Interface Designed for Multimedia. Academic Press, 1999.

[58] Angela Doufexi, David Redmill, David Bull, and Andrew Nix. MPEG-2 Video Transmission Using the Hiperlan/2 WLAN standard. In *IEEE Transactions on Consummer Electronics*, volume 47, August 2001.

[59] Ben Brown and Mark Beach. Open infotainment services in radio interconnected systems (osiris): Project plan, 3cr/d8. Technical report, 3C Research Program, September 2003.

[60] Wireless Cameras and Audio-Visual Seamless Networking. `http://www.ist-wcam.org`, 2004-2005.

[61] `http://www.wi-fi.org`.

[62] Wi-Fi Alliance. Wi-Fi CERTIFIED for WMM - Support for Multimedia Applications with Quality of Service in Wi-Fi Networks. Technical report, Wi-Fi Alliance, September 2004.

[63] Fred Halsall. *Data Communications, Computer Networks and Open Systems - 4th Edition.* Addison - Wesley, 1996.

[64] J. Postel. Request For Comments - RFC 768 User Datagram Protocol, August 1980.

[65] Zhenping Zuo. In-building Wireless LANs. `http://www.cse.ohio-state.edu/~jain/cis788-99/wireless_lans`, 2000.

[66] Edward C. Prem. Wireless Local Area Network. `http://www.ictp.trieste.it/~radionet/2000_school/lectures/carlo/802.11/wireless_lans`, 1997.

[67] ETSI Hiperlan Type 2; Broadband Radio Access Network (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Function ETSI (DTS/BRAN0020004-1), December 1999.

[68] Martin Johnsson. HiperLAN/2 - The Broadban Radio Transmission Technology Operating in the 5GHz Frequency Band. `http://www.hiperlan2.com`, 1999.

[69] Jamshid Khun-Jush, Peter Schramm, Udo Wachsmann, and Fabian Wenger. Structure and performance of the Hiperlan/2 Physical Layer. In *IEEE Vehicular Technology Conference (VTC)*, Amsterdam, September 1999.

[70] Bernard Sklar. Rayleigh Fading Channels in Mobile Digital Communication Systems, Part 1: Characterisation. In *IEEE Communication Magazine*, July 1997.

[71] J.H Stott. The How and Why of COFDM. Technical report, BBC, 1998.

[72] R. Van Nee, G.Awater, M.Morikura, H.Takanashi, M.Webster, and K.Halford. New High-Rate Wireless LAN Standards. In *IEEE Communication Magazine*, December 1999.

[73] T.Weiss, A.Krohn, and F.Jondral. Synchronization Algorithms and Preamble Concepts for Spectrum Pooling Systems. In *Mobile and Wireless Communications Summit (IST - MWC)*, Aveiro, June 2003.

[74] Z. Lin, G. Malmgren, and J. Torsner. System Performance Analysis of Link Adaptation in HiperLAN Type 2. In *IEEE Vehicular Technology Conference (VTC)*, Boston, October 2000.

[75] Qiang Ni, Lamia Rondhani, and Thierry Turletti. A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN. In Wiley, editor, *Journal of Wireless Communications and Mobile Computing*, volume 4, 2004.

[76] Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix, and David Bull. Robust Video Transmission over Wireless LANs. In *IEEE Transactions on Vehicular Technology*, Submitted, under review 2004.

[77] Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix, and David Bull. Packetisation Strategies for Enhanced Video Transmission over Wireless LANs. In *IEEE Packet Video workshop (PV)*, Irvine, December 2004.

[78] Raphael Rom and Moshe Sidi. *Multiple Access Protocols - Performance and Analysis.* Springer-Verlag, 1989.

[79] Joao L.Sobrinho and A.S Krishnakumar. Real-Time Traffic over the Ieee 802.11 Medium Access Control Layer. Technical report, Bell Labs - Technical Journal, Automn 1996.

[80] Pierre Ferré, Angela Doufexi, Andrew Nix, and David Bull. Throughput Analysis of the IEEE 802.11 and IEEE 802.11e MAC. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, March 2004.

[81] Magis Networks. IEEE 802.11e/a Throughput Analysis, E10282. `http://www.magisnetworks.com`, 2003.

[82] Jason Liu, David M.Nicol, L.Felipe Perrone, and Michael Liljenstam. Toward High Performance Modeling of the 802.11 Wireless Protocol. In *Winter Simulation Conference*, Arlington, December 2001.

[83] Kin K. Leung, Bruce McNair, Leonard J.Cimini Jr., and Jack H. Winters. Outdoor IEEE 802.11 Cellular Networks: MAC Protocol Design and Performance. In *IEEE International Conference on Communications (ICC)*, New-York, May 2002.

[84] Hiroyuki Yomo, Shyam S. Chakraborty, and Ramjee Prasad. IEEE802.11 WLAN with Packet Combining. In *International Conference on Computer and Devices for Communication*, India, January 2004.

[85] Stefen Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hertz, and Lothar Stibor. IEEE802.11e Wireless LAN for Quality of Service. In *European Wireless*, Florence, February 2002.

[86] Giuseppe Anastasi, Luciano Lenzini, Enzo Mingozzi, Andreas Hettich, and Andreas Kramling. MAC Protocols for Wideband Wireless Local Access: Evolution Toward Wireless ATM. In *IEEE Personal Communications*, October 1998.

[87] Giuseppe Bianchi. IEEE 802.11 - Saturation Throughput Analysis. In *IEEE Communcations Letters*, volume 2, December 1998.

[88] Zoran Hadzi-Velkov and Boris Spasenovski. Saturation Throughput - Delay Analysis of the IEEE802.11 DCF in Fading Channel. In *IEEE International Conference on Communications (ICC)*, Anchorage, May 2003.

[89] P.Chatzimisios, V.Vitsas, and A.C.Boucouvalas. Throughput and Delay Analysis of the IEEE802.11 Protocol. In *International Workshop on Network Appliances (IWNA)*, Liverpool, October 2002.

[90] Roger CJ. Clarke. *Digital Compression of Still Images and Video.* Academic Pres, 1995.

[91] Mohamed Ghanbari. *Video Coding, an Introduction to Standard Codecs*, volume 42. IEE Telecommunication Series, 1999.

[92] Arun N. Netravali and Barry G. Haskell. *Digital Pictures, Representation, Compression and Standards, Second Edition.* Plenum Press, New-York, 1994.

[93] Simon Haykin. *Communication Systems - 3rd Edition.* Wyley, 1994.

[94] A.M. Tekalp. *Digital Video Processing.* Prentice Hall, 1995.

[95] ITU-R Recommendation BT.500-11. Methodology for the Subjective Assessment of the Quality of Television Pictures, 2002.

[96] B.G Haskell, A.Puri, and A.N Netravali. *Video: An Introduction to MPEG2 - Digital Multimedia Series.* Chapman and Hall, 1997.

[97] ISO/IEC 1117-2. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbits/s, November 1991.

[98] ISO/IEC 13818-2. Generic Coding of Moving Pictures and Associated Audio: Video Recommendation H.262.0, November 1994.

[99] ISO/IEC 13818-1. Generic Coding of Moving Pictures and Associated Audio: Systems Recommendation H.262.0, 1996.

[100] P.A Sargison. MPEG-2: Overview of the System Layer. Technical report, BBC, 1996.

[101] Use of MPEG2 System Streams in Digital Imagery Systems - MISB RP 0101. Technical report, Motion Imagery Standards Boards, March 2001.

[102] Yao Wang and Qin-Fan Zhu. Error Control and Concealment for Video Communication: A Review. In *Proceedings of the IEEE*, volume 86, May 1998.

[103] David W. Redmill and N.G Kingsbury. The EREC: An Efficient Error-Resilient Technique for Coding Variable-Length Blocks of Data. In *IEEE Image Processing*, April 1996.

[104] W.M. Lam, A.R Reibman, and B.Liu. Recovery of Lost or Erroneously Received Motion Vectors. In *IEEE International Conference On Accoustics, Speech and Signal Processing (ICASSP)*, Minneapolis, April 1993.

[105] Wonki Kim and Jechang Jeong. A Temporal Error Concealment Technique Using Motion Adaptionve Boundary Matching Algorithm. In *International Conference on Electronic, Information and Communications*, Hanoi, August 2004.

[106] Luigi Atzori, Francesco G.B de Natale, and Cristina Perra. A Spatial-Temporal Concealment Technique Using Boundary Matching Algorithm and Mesh-Based Warping (BMA-MBW. In *IEEE Transactions on Multimedia*, September 2001.

[107] Jae-Young Pyun, Jun suk Lee, Jin-Woo Jeong, Jae-Hwan Jeong, and Sung-Jae Ko. Robust Error Concealment for Visual Communications in Burst-Packet-Loss Networks. In *IEEE Transactions on Consummer Electronics*, volume 49, November 2003.

[108] ISO/IEC 14496-2. Coding of Audio-Visual Objects - Part 2: Visual, 2001.

[109] Touradj Ebrahimi and Caspar Horne. MPEG-4 Natural Video Coding. In *Elsevier Journal on Signal Processing: Image Communication*, volume 15, 2000.

[110] Guy Côté, Bernd Erol, Michale Galland, and Faouzi Kossentini. H.263+: Video Coding at Low Bit Rates. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 8, November 1998.

[111] Iain E.G Richardson. *H.264 and MPEG-4 Video Compression - Video Coding for Next-Generation Multimedia.* Wiley, 2003.

[112] ITU-T Recommendation H.261. Video Codec for Audiovisual Services at px64kbits/s, 1990.

[113] ITU-T Recommendation H.263. Video Coding for Low Bit Rate Communication, May 1996.

[114] ITU-T Recommendation H.263 version 2 (H.263+). Video Coding for Low Bit Rate Communication, December 1999.

[115] Guy Côté and Lowelle Winger. Recent advances in Video Compression Standards. In *IEEE Canadian Review*, Spring 2002.

[116] Ralf Schafer, Thomas Wiegand, and Heiko Schwarz. The Emerging H.264/AVC Standard. Technical report, EBU Technical Review, January 2003.

[117] Ajay Luthra and Pankaj Topiwala. Overview of the H.264/AVC Video Coding Standard. In Andrew G. Tescher, editor, *Proceedings of SPIE: Applications of Digital Image Processing XXVI*, volume 5203, 2003.

[118] M.Madhi Ghandi and Mohammad Ghanbari. The H.264/AVC Video Coding Standard for the Next Generation Multimedia Communication. In *IAEEE Journal.*

[119] Nejat Kamaci and Yucel Altunbasak. Performance Comparison of the Emerging H.264 Video Coding Standard With the Existing Standards. In *IEEE International Conference on Multimedia and Expo (ICME)*, Baltimore, July 2003.

[120] Till Halbach. The H.264 Video Compression Standard. In *Norvegian Signal Processing Symposium - NORSIG*, Bergen, October 2003.

[121] C.M. Calagate and M.P Malumbres. Testing the H.264 Error Resilience on Wireless Ad-Hoc Networks. In *European Association for Signal Processing (EURASIP), Video/Image Processing and Multimedia Communications*, Zagreb, July 2003.

[122] Till Halbach and Steffen Olsen. Error Robustness Evaluation of H.264/MPEG4 AVC. In *Visual Communications and Image Processing Conference (VCIP)*, San Jose, January 2004.

[123] Marta Karczewicz and Ragip Kurceren. The SP- and SI-Frames Design for H.264/AVC. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 13, July 2003.

[124] Dong Tian, Miska M. Hannuksela, Ye-Kui Wang, and Moncef Gabbouj. Error Resilient Video Coding Techniques Using Spare Pictures. In *IEEE Packet Video workshop (PV)*, Nantes, April 2003.

[125] Thomas Stockhammer, Thomas Wiegand, Tobias Oelbaum, and Florian Obermeier. Video Coding and Transport Layer Techniques For H.264/AVC-Based Transmission Over Packet-Lossy Networks. In *IEEE International Conference on Image Processing (ICIP)*, Barcelona, September 2003.

[126] Gary.J Sullivan, Thomas Wiegand, and Thomas Stockhammer. Using the Dratf H.26L Video Coding Standard for Mobile Applications. In *IEEE International Conference on Image Processing (ICIP)*, Thessaloniki, October 2001.

[127] Ivaylo Haratcherev and Koen Langendoen. Hybrid Rate Control for IEEE802.11. In *International Workshop on Mobile Multimedia Comminucation (MoMuc)*, Munich, October 2003.

[128] Thomas Stockhammer, Miska M. Hannuksela, and Stephan Wenger. H.26L/JVT Coding Network Abstraction Layer and IP-Based Transport. In *IEEE International Conference on Image Processing (ICIP)*, Rochester, September 2002.

[129] Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix, and David Bull. Cross Layer Adaptation for Video Transmission over Wireless LANs. In *IEEE Transactions on Circuits and Systems for Video Technology*, Submitted, under review 2004.

[130] Stephan Wenger, Miska M. Hannuksela, Thomas Stockammer, M. Westerlund, and D.Singer. Request For Comments - RFC 3984 - RTP Payload Format for H.264 Video Format, February 2005.

[131] H. Shchulzrinnel. Request For Comments - RFC 3550 - RTP: a Transport Protocol for Real-Time Applications, July 2003.

[132] Karsten Sühring. H.264/AVC Software Coordination. `http://bs.hhi.de/~suehring/tml/`.

[133] Viktor Varsa, Miska M. Hannuksela, Ari Hourunranta, and Ye-Kui Wang. Non-Normative Error Concealment algorithms - Document VCEG-N62. Technical report, ITU - Telecommunications Standardisation Sector, Video Coding Expert Group (VCEG), Santa Barbara, September 2001.

[134] S.Krishnamachari, M.Van der Schaar, S.Choi, and X.Xu. Video Streaming over Wireless LANs: A Cross Layer Approach. In *IEEE Packet Video workshop (PV)*, Nantes, April 2003.

[135] Nick Flaherty. Wi-fi Heads for the Home. In *IEE Communication Engineer*, August 2003.

[136] D.Dardari, M.G.Martini, M.Milantoni, and M.Chiani. MPEG-4 Video transmission in the 5GHz Band Through an Adaptive OFDM Wireless Scheme. In *Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Lisbon, September 2002.

[137] Vladimir Stankovic, Raouf Hamzaoui, and Zixiang Xiong. Live Video Streaming over Packet Networks and Wireless Channels. In *IEEE Packet Video workshop (PV)*, Nantes, April 2003.

[138] Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix, and David Bull. Enhanced Video Streaming over COFDM based Wireless LANs using combined Space Time Block Coding and Reed Solomon Concatenated Coding. In *IEEE Vehicular Technology Conference (VTC)*, Milan, May 2004.

[139] Peter Sweeney. *Error Coding Control.* Wiley, 2002.

[140] Alain Glavieux and Michel Joindot. *Communications Numeriques: Introdution*, chapter VIII: Code Correcteur d'Erruers. Masson, 1996.

[141] ETSI DVB-S; Digital Video Broadcasting (DVB), Framing Structure, Channel Coding and Modulation for 11/12GHz Satellite Services, ETSI EN 300 421, v1.1.2, August 1997.

[142] M.Alamouti. A Simple Transmit Diversity Technique for Wireless communications. In *IEEE Journal on Selected Areas in Communications*, volume 16, October 1998.

[143] V.Tarokh, H.Jafarkhani, and A.R.Calderbank. Space Time Block Coding for Wireless communications: Performance Results. In *IEEE Journal on Selected Areas in Communications*, volume 17, March 1999.

[144] Angela Doufexi, Simon Armour, Andrew Nix, and Mark Beach. Throughput Enhancement of OFDM-based WLANs Using Space Time Block Codes and Time Domain Least Squares Channel Estimation. In *European Personal Mobile Communications Conference (EPMCC)*, Glasgow, April 2003.

[145] Angela Doufexi, Aranxta Prado Miguelez, Simon Armour, Andrew Nix, and Mark Beach. Use of Space Time Block Codes and Spatial Multiplexing using TDLSChannel Estimation to Enhance the Throughput of OFDM based WLANs. In *IEEE Vehicular Technology Conference (VTC)*, Jeju, Spring 2003.

[146] Sunghyun Choi. IEEE802.11e MAC-Level FEC performance Evaluation and Enhancement. In *IEEE Global Telecommunications Conference (GlobeCom)*, Taipei, November 2002.

[147] Syed Aon Mujtaba. IEEE P802.11; TGn Sync Proposal Technical Specification for IEEE 802.11 TGn- doc.: IEEE 802.11 -04/0889r6, May 2005.

[148] Ka Yeung. 802.11a Modeling and MAC Enhancements for High Speed Rate Adaptive Networks - cs 219. Technical report, University of California, Los Angeles, December 2002.

[149] Xiofeng Xu, Mihaela van der Schaar, Santhana Krishnamachari, Sunghyun Choi, and Yao Wang. Adaptive Error Control for Fine-Granularity-Scalability Video Coding over IEEE 802.11 Wireless LANS. In *IEEE International Conference on Multimedia and Expo (ICME)*, Baltimore, July 2003.

[150] Yang Xiao. Concatenation and Piggyback Mechanisms for the IEEE802.11 MAC. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, March 2004.

[151] T. Turletti et al. Request For Comments - RFC 2032 - RTP Payload Format for H.261 Video Streams, October 1996.

[152] C. Zhu. Request For Comments - RFC 2190 - RTP Payload Format for H.263 Video Streams, September 1997.

[153] C. Bormann et al. Request For Comments - RFC 2429 - RTP Payload Format for 1998 Version of the ITU-T Rec. H.263+, October 1998.

[154] J. Postel. Request For Comments - RFC 793 Transmission Control Protocol, September 1981.

[155] L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsson, and G.Fairhurst. Request For Comments - RFC 3528 The Lightweigth User Datagram Protocol, July 2004.

[156] Lars-Ake Larzon, Mikael Degermark, and Stephen Pink. UDP-Lite for Real-Time Multimedia Applications. In *IEEE International Conference on Communications (ICC)*, Vancouver, June 1999.

[157] S. Deering. Request For Comments - RFC 2460 Internet Protocol, Version 6 IPv6, September 1981.

[158] Pierre Ferré, Dimitris Agrafiotis, Tuan Kian Chiew, Angela Doufexi, Andrew Nix, and David Bull. Packet Loss Modelling for H.264 Video Transmission over IEEE 802.11 Wireless LANs. In *Workshop on Image Analysis for Multimedia Interactive Service (WIAMIS)*, Montreux, April 2005.

[159] Youngsoo Kim, Sunghyun Choi, Hyosun Hwang, and Kyunghun Jang. Throughput Enhancement via Frame Aggregation. In *IEEE Vehicular Technology Conference (VTC)*, Los Angeles, Fall 2004.

[160] Christine Guillemot and Paul Christ. Joint source-channel coding as an element of a qos framework for 4G wireless multimedia. In *Elsevier Journal on Computer Communications*, volume 27, May 2004.

[161] C. Bormann et al. Request For Comments - RFC 3095 - RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed, July 2001.

[162] Mihaela van der Schaar and Sai Shankar. Cross-Layer Wireless Multimedia Transmission: Challenges, Principles and New Paradigms. In *IEEE Wireless Communications*, August 2005.

[163] Ivaylo Haratcherev, Keon Langendoen, Inald Lagendijk, and Henk Sips. D3.16: Application-directed automatic 802.11 rate control. Technical report, GigaMobile Project, TU Delf, 2002.

[164] Ivaylo Haratcherev, Koen Langendoen, Reginald Lagendijk, and Henk Sips. SNR-based Rate Control in WaveLAN. In *ASCI 2004 Conference*, Port Zelande, June 2004.

[165] Pierre Ferré, Angela Doufexi, James Chung-How, Andrew Nix, and David Bull. Link Adaptation for Video Transmission over COFDM basedWLANs. In *IEEE Symposium on Communications and Vehicular Technology (SCVT)*, Eindhoven, November 2003.

[166] Hua Zhu, Ming Li, Imrich Chlantac, and B. Prabhakaran. A Survey of Quality of Service in IEEE 802.11e Networks. In *IEEE Wireless Communications*, August 2004.

[167] A.J. van der Vegt. Auto Rate Fall Back Algorithm for IEEE 802.11a Standard. Technical report, High Performance Computing Group - Faculty of Physics and Astronomy - University of Utrech, 2002.

[168] Mohammad Hossein Manshaie, Thierry Turletti, and Marwan Krunz. A Media-Orientated Transmission Mode Selection in 802.11 Wireless LANs. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, March 2004.

[169] Daji Qiao, Sunghyun Choi, Amit Jain, and Kang G. Shin. MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h. In *Mobicom*, San Diego, September 2003.

[170] Daji Qiao, Sughyun Choi, Amit Jain, and Kand G. Shin. Adaptive Transmit Power in Control in IEEE 802.11a Wireless LANs. In *IEEE Vehicular Technology Conference (VTC)*, Jeju, Spring 2003.

[171] Dilip Krishnaswamy. Network-Assisted Link-Adaptation with Power Control and Channel Re-Assignment in Wireless Networks. In *3G Wireless Conference*, San Fransisco, May 2002.

[172] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Mobicom*, Rome, July 2001.

[173] Song Ci and Hamid Sharif. A Variable Data Rate Scheme to Enhance Throughput Performance of Wireless LANs. In *Symposium on Communication System, Network and Digital Signal Processing (CSNDSP)*, Staffordshire University, July 2002.

[174] Wing Ho Yuen, Heung-No Lee, and Timothy D. Andersen. A Simple and Effective Cross Layer Networking Sytem for Mobile A-Hoc Networks. In *Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Lisbon, September 2002.

[175] Ceilidh Hoffman, Mohammad Hossein Manshaie, and Thierry Turletti. CLARA: Closed-Loop Adaptive Rate Allocation for IEEE 802.11 Wireless LANs. In *Conference on Wireless Networks, Communications, and Mobile Computing*, Hawaii, June 2005.

[176] Ivaylo Haratcherev and Koen Langendoen. Hybrid Rate Control for IEEE802.11. In *ACM International Workshop on Mobility Management and Wireless Access (MobiWac)*, Philadelphia, September 2004.

[177] Ivaylo Haratcherev, Jacco Taal, Koen Langendoen, Reginald Lagendijk, and Henk Sips. Automatic IEEE 802.11 Rate Control for Streaming Applications. In Wiley, editor, *Journal of Wireless Communications and Mobile Computing*, volume To appear, 2005.

[178] Ivaylo Haratcherev, Jacco Taal, Koen Langendoen, , Reginal L. Lagendijk, and Henk Sips. Adaptive End-to-End Optimisation of Mobile Video Streaming Using QoS Negociation. In *IEEE International Symposium on Circuits And Systems (ISCAS)*, Kobe, May 2005.

[179] Jacco R. Taal, Koen Langendoen, Arjen van der Schaaf, Hylke W. van Dijk, and Reginal L. Lagendijk. Adaptive End-to-End Optimisation of Mobile Video Streaming Using QoS Negociation. In *IEEE International Symposium on Circuits And Systems (ISCAS)*, Scottsdale, May 2002.

[180] S. Saeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic Media Access for Multirate Ad-Hoc Networks. In *Mobicom*, Atlanta, September 2002.

[181] Severine Catreux, Vinko Erceg, David Gesbert, and Robert W.Heath Jr. Adaptive Modulation and MIMO Coding for Broadbad Wireless Data Networks. In *IEEE Communication Magazine*, June 2002.

[182] Tuan Kiang Chiew, Pierre Ferré, Dimitris Agrafiotis, Araceli Molina, Andrew Nix, and David Bull. Cross-Layer WLAN Measurement and Link Analysis for Low Latency Error Resilient Wireless Video Transmission. In *IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, January 2005.

[183] Nada Golmie and Frederci Mouveaux. Interference in the 2.4GHz ISM Band: Impact on the Bluetooth Access Control Performance. In *IEEE International Conference on Computer Communication (INFOCOM)*, Anchorage, April 2000.

[184] IEEE Std for Information Technology 802.151, Part 15.1: Wireless Medium Acess Control MAC and Physical Layer Specifications for Wireless Personal Area Networks (WPANs), June 2002.

[185] Bluetooth Special Interest Group (SIG) Ericsson. Bluetooth 2: The high-rate extension of Bluetooth. `http://www.bluetooth.com`, September 2000.

[186] The Bluetooth Standard Version 1.0. `http://www.bluetooth.com`, July 1999.

[187] Jaap Haartsen. BLUETOOTH - The universal radio interface for ad hoc, wireless connectivity. Technical report, Ericsson, June 1998.

[188] Alain Glavieux. Introduction a la Theorie de l'Information et du Codage de Canal, 1996-1997.

[189] Dongyang Chen, Daqing Gu, and Jinyun Zhang. Supporting Real-time Traffic wiht QoS in IEEE802.11e Based Home Networks. In *IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, January 2004.

[190] Daqing Gu and Jinyun Zhang. Evaluation of the EDCF Mechanism for QoS in IEEE802.11 Wireless Networks. In *World Wireless Congress*, San Fransisco, May 2003.

[191] Joshua Wall and Jamil Y.Khan. An ARQ Enhancement with QoS Support for the 802.11 MAC Protocol. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, March 2004.

[192] Gabriella Convertino. 802.11e Overview. Technical report, Advanced System Technology - Audio Video over Wireless LAN Group.

[193] Carlos T.Calafate, Juma Carlos, Pietro Manzoni, and Manuel P.Malumbres. Achieving Enhanced Performance in MANETs using IEEE 802.11e. In *XV Jornadas de Paralelismo*, Almeria, September 2004.